



# Maximization of Attractiveness EV Tourist Routes

Joanna Karbowska-Chilinska, Pawel Zabielski

## ► To cite this version:

Joanna Karbowska-Chilinska, Pawel Zabielski. Maximization of Attractiveness EV Tourist Routes. 16th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Jun 2017, Bialystok, Poland. pp.514-525, 10.1007/978-3-319-59105-6\_44 . hal-01656202

**HAL Id: hal-01656202**

**<https://inria.hal.science/hal-01656202>**

Submitted on 5 Dec 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Maximization of attractiveness EV tourist routes

Joanna Karbowska-Chilinska (ORCID 0000-0001-5241-7930) and Pawel Zabieliski (ORCID 0000-0002-0126-1849)

Bialystok University of Technology, Faculty of Computer Science, Poland

**Abstract.** This paper presents model and an algorithmic approach for the problem of generation optimal tourist route for electric vehicles (EVs). In the discussed problem a starting and a final point of a route are EV charging stations where tourist could charge the battery and then continue a journey. The main objective is to select to the route points of interests (POIs) which maximizing tourist attractiveness. Furthermore maximum length of the route is limited by the number of kilometers that the car can travel on a single battery charge. The model applied by us is the graph routing problem named as the Orienteering Problem with Time Windows (OPTW). In OPTW each location has positive score and a specific time interval in which a location can be visited. The solution of OPTW is a route (from the given starting to the ending point) with a fixed limit of length including a subset of locations. Moreover the route maximizes the total score of the locations visited in the predefined time intervals. As a solution we present the evolutionary algorithm with combines path relinking method instead crossover. Computational experiments are conducted on realistic database POIs and EV charging stations of Podlasie region in Poland. Tests results and execution time of the algorithm shows that the described solution could be a part of EV software module with generates the most interesting route.

**Keywords:** Orienteering Problem with Time Windows · Tourist Trip Generation · Electric Vehicles · Evolutionary Algorithm

## 1 Introduction

Nowadays carbon dioxide emissions have rapidly increased due to cities development and population growth. Transport has a high negative impact, on the environment degradation. In this reason in last years Green Logistics (GL) has received increasing attention in business as well as in a research field [3]. The main goals of GL are: maximum utilization of means of transport, development of green intelligent transportation systems which reduces the energy consumption, promotion of alternative fuels and vehicles with electric engines. Environmental aspects has been included to the optimization problem of vehicle routing problems (VRP) and also green vehicle routing problem is developed (G-VRP) [3]. In modern transportation system the electrical vehicles (EVs) are beneficial to the environment. EVs are more silent in operation and emission free compared

to gasoline powered cars. However the driving range of a car depends largely on the capacity of the battery. It is important constraint which affect into routing planning for EVs. Batteries need to be charged a significant amount of time extending the route time. Literature describes some problems such as Electric Vehicle Scheduling Problem (E-VSP) [23], Electric Vehicle Routing Problem (EVRP) [1], Electric Vehicles Routing problem with Time Windows and Recharging Stations(E-VRPTW) [15] which solutions minimizing total traveled distance and give possibility of recharging batteries.

In many touristic regions when clear air and national parks are important e.g. US [5] (California, Arizona, Orlando), Europe (Goms and Haslital region of the Swiss Alps [4]) the Electric Vehicle Tourism is popularized. Especially EV rental cars services are developed in those touristic regions. Tourist that use EV are satisfied when they could choose the most interesting POIs for each trip day and have not to care about charging the battery. It is comfortable if the battery charging can take place during e.g. an accommodation or the other break in a tour. The opening and closing time assigned to each POIs is additional constraint considered in a tour scheduling.

To the best our knowledge any web, mobile and cars applications (e.g. rang assistant in BMW i3, EV TripPlanner for Tesla S users [6]) do not select the set of POIs maximizing tourist satisfaction and taking into account the trip constraints such as: driving range of a car battery as well as POIs visiting hours.

In this regard, this article presents the suitable model for the described problem. Moreover the effective evolutionary algorithm with path relinking instead crossover developed by us in [12] is used to generate the most satisfying tourist trip for EVs user with the aforementioned limitations. The graph optimisation routing problem called Orienteering Problem with Time Windows (OPTW) [20] is used by us as the model. The considered graph have vertices with assigned score/ profit which was interpreted by us as a vertex attractiveness.

An extension of OPTW is Team Orienteering Problem with Time Windows (TOPTW) where the given number of optimal routes (the parameter  $m$ ) are generated, each one not exceeded the same constraint. Methods for the TOPTW could also be applied to the OPTW because the solution OPTW is only one optimal route ( $m=1$ ). Both these problems belongs to the class NP- hard [22]. In literature there are many meta-heuristic approaches giving satisfactory solutions e. g. local search methods [21], tabu search [19], ant colony optimisation [16], variable neighbour search [14], evolutionary inspired algorithms [17].

The remainder of the paper is organized as follows. The OPTW problem definition and its transformation to the problem of generation optimal tourist routes of EVs are presented in Section 2. In Section 3, we describe the concept of our evolutionary algorithm for solving the problem. The computational experiments run on realistic database POIs and EV charging stations of Podlasie region in Poland are discussed in Section 4. Finally the paper is concluded in Section 5. In this section we also lay the groundwork for further research.

## 2 The Problem Definition

In this work the optimization of an EV tourist route is modeled by the known graph problem called OPTW [10], [22]. In OPTW a set of the fixed locations is given. Each location  $i$  has a positive score  $S_i$ , which is added after the visit of the location and a visiting time  $T_i$  as well as a time window  $[O_i, C_i]$ , where  $O_i$  and  $C_i$  denote the opening and closing time, respectively. Service of a location must begin and end within this interval. It is permissible to wait before the opening time in order to visit profitable locations and maximize the total score. Each edge between locations  $i$  and  $j$  has a fixed cost  $t_{ij}$  which is interpreted as distance needed to travel between locations. The OPTW goal is to determine a single route from the fixed starting to the ending point, that visits some of vertices within the fixed time windows and maximizes the total score. Moreover, the total cost of the edges on the route must be less than the threshold  $t_{max}$ , and any vertex on the path can only be visited once. A feasible solution not violate any time window constraint on the fixed travel time limit (or travel length) of the route.

In the case of the optimization of a tourist route of EV the set  $A$  of  $n$  POI(s) and the set  $B$  of  $m$  EV charging stations are given. Each POI  $i$  has assigned interval  $[O_i, C_i]$  in which a visit can last and the fixed visiting time  $T_i$ . The distances between each pair of POIs and each POI and EV charging station are denoted by  $t_{ij}$  for  $i, j \in \{1, \dots, n + m\}$ . Each POI  $i$  has a score  $S_i \geq 0$  interpreted as its attractiveness. The EV charging stations have profits equal to zero. Moreover  $t_{max}$ - the number of kilometers that the consider model of car can travel on a single battery charge is known. The starting and ending points, denoted by  $s$  and  $e$  respectively are selected among the points from the set  $B$  and their distance does not exceed the limit  $t_{max}$ . The optimization goal is the generation a route from starting to the ending point that maximizes the total collected score (connected with attractiveness of the route), each POI is visited at most once in the predefined time interval and the length of the route does not exceed the limit  $t_{max}$ . The problem can be formulate as a mixed integer problem with the decision variable:  $x_{ij} = 1$  if the direct link between  $i$  and  $j$  is included in a route, and  $x_{ij} = 0$  otherwise. Moreover  $start_i$  denotes the start of service time at vertex  $i$  and  $M$  is a large constant.

The objective function (1) maximizes the total collected profit of the route:

$$\max \sum_{i=1}^{n+m-1} \sum_{j=2}^{n+m} S_i x_{ij}. \quad (1)$$

The constraint in (2) guarantees that the path starts at vertex 1 and ends at vertex  $n + m$ :

$$\sum_{j=2}^{n+m} x_{1j} = \sum_{i=1}^{n+m-1} x_{in+m} = 1. \quad (2)$$

Constraint (3) requires that there may be at most one visit to any vertex:

$$\sum_{i=1}^{n+m-1} x_{ik} = \sum_{j=2}^{n+m} x_{kj} \leq 1 \quad \forall k \in \{2, \dots, n+m-1\}. \quad (3)$$

The constraint in (4) ensures that the distance of the route is limited by  $t_{max}$ :

$$\sum_{i=1}^{n+m-1} \sum_{j=2}^{n+m} t_{ij} x_{ij} \leq t_{max}. \quad (4)$$

Constraint (5) ensures the timeline of the route:

$$start_i + T_i + t_{ij} - start_j \leq M \cdot (1 - x_{ij}). \quad (5)$$

The start of the service is restricted by a time window as in (6):

$$O_i \leq start_i \leq C_i \quad \forall i = 1, \dots, n+m. \quad (6)$$

### 3 Evolutionary Algorithm

To solve the presented problem, the evolutionary algorithm (EA) is used, previously developed by us to solve OPTW [12]. The EA for OPTW gives about 2% better results in comparison to other metaheuristics like iterated local search or greedy randomized adaptive search procedure [12]. In this reason we decide to use EA also to solve the presented in this paper problem. In the algorithm instead of the crossover a path relinking (PR) strategy is used. PR starts from selecting initial and guiding solutions to represent the starting and the ending points of the route [7]. Attributes from the guiding solution are gradually introduced into the intermediate solutions, so that these solutions contain less characteristics from the initial solution and more from the guiding solution. As a result, the selective moves between initial and guiding solutions provide the better routes than the input routes. In the literature PR significantly improves the different metaheuristics solving different optimization problems e.g. greedy randomized adaptive search procedure (GRASP) [2], variable neighbor search (VNS) [18], genetic algorithms (GA) [8] and tabu search (TS) [9].

Let  $N_g$  denotes a number of the algorithm generations,  $P_{size}$  describes the number of routes in the initial population.

The basic structure of the EA is given as follows:

```

compute initial P_size routes;
iter=0; not_imp=0; stop= false
while (iter< Ng) and (stop =false) do
    iter++;
    evaluate the fitness value of each route;
    make a tournament grouping selection;
    perform path relinking;

```

```

    perform mutation;
    if no improvements in an attractiveness after 100 iterations
    then stop= true;

od
return the route with the highest profit value;

```

The subsequent steps of the EA are described in detail in the following subsections.

### 3.1 Initialization

In our approach a route is coded as a sequence of POIs but the starting ( $s$ ) and ending ( $e$ ) points are the fixed among EV charging stations. First the route is initialized by  $s$  and  $e$  stations. Then the following values are assigned sequentially to a POI selected to route:  $arrival_i$  - arrival time at POI  $i$ ,  $wait_i$  - waiting time, if the arrival at a POI  $i$  is before opening time,  $start_i$  and  $end_i$  - starting and ending service time at POI  $i$ . Moreover, the maximum time the a visit  $i$  can be delayed without making other visits infeasible is calculated for each POI in the route as follows [21]:

$$MaxShift_i = Min(C_i - start_i - T_i, wait_{i+1} + MaxShift_{i+1}) \quad (7)$$

Let POI  $l$  be the predecessor of charging station  $e$  in the route. In the subsequent steps a set of POIs is included. Each location  $v$  from this set is adjacent to location  $l$  and the station  $e$  and will satisfy the following conditions after insertion: (a)  $start_v$  and  $end_v$  are within the range  $[O_v, C_v]$ ; (b) the locations after  $v$  could be visited in the route; and (c) the current travel length of the route does not exceed the given  $t_{max}$ . A random location  $v$  is chosen from this set. The values  $arrival_v$ ,  $wait_v$ ,  $start_v$  and  $end_v$  are calculated and the location  $v$  is inserted. After the insertion, the values  $arrival_e$ ,  $wait_e$ ,  $start_e$  and  $end_e$  are updated. Moreover, for each POI in the tour (from vertex  $e$  to  $s$ ) the  $MaxShift$  value is updated as well. The route generation is continued for as long as POIs that have not been included are present and  $t_{max}$  is not exceeded.

### 3.2 Selection

We use tournament grouping selection used by us in [24] and in [11]. This selection yields better adapted individuals than standard tournament selection. In this method a set of  $P_{size}$  routes is divided into  $k$  groups and the tournaments are carried out sequentially in each of the groups.  $t_{size}$  routes are removed from the group, the chromosome with the highest value for the fitness function in the form:

$$\frac{TotalProfit^3}{TravelTime} \quad (8)$$

is copied to the next population, and the  $t_{size}$  previously chosen individuals are returned to the old group.  $TotalProfit$  denotes sum of POI's profits in the route and  $TravelTime$  is the time connected with the visit of the POI's as well as length of the route. After repetition of  $P_{size}/k$  selection from the group currently analyzed,  $P_{size}/k$  routes are chosen for a new population. Finally, when this step has been repeated in each of the remaining groups, a new population is created, containing  $P_{size}$  routes.

### 3.3 Path Relinking

In this paper PR is used by us as alternative to the crossover. The method was developed by us with good results on benchmarks in the case of OPTW solution in [12]. Let  $V_{R_1-R_2}$  be the set of POIs present in a route  $R_1$  and not presented in  $R_2$ , and let  $V_{R_2-R_1}$  denotes the set of POIs present in  $R_2$  and not in  $R_1$ . During  $PR(R_1, R_2)$  we attempt to insert POIs from  $V_{R_2-R_1}$  into  $R_1$  in the best possible positions. The total consumption time associated with inserting a POI  $j$  between  $i$  and  $k$  is calculated in the following way:

$$Shift_j = t_{ij} + wait_j + T_j + t_{jk} - t_{ik}. \quad (9)$$

In addition, we check whether the shift resulting from the new insertion exceeds the constraints associated with the previously calculated  $wait$  and  $MaxShift$  values for the vertices located directly after the newly inserted one. If the shift exceeds the constraints the vertices from  $V_{R_1-R_2}$  are removed to restore the possibility of inserting new POIs. For each vertex  $u$  from this set a ratio is calculated as follows:

$$RemovalRatio = \frac{(p_u)^2}{end_u - arrive_u}. \quad (10)$$

After this computation the POI with the smallest value for  $RemovalRatio$  is removed. This removal is repeated until we can insert some vertices into the path. Finally the vertex  $u$  with the highest value:

$$InsertionRatio = \frac{(p_u)^2}{Shift_u} \quad (11)$$

is selected for insertion if not exceed the mentioned constraints. After  $u$  is inserted, the values of  $arrive_u$ ,  $wait_u$ ,  $start_u$  and  $end_u$  are calculated. For each location in the route after  $u$  the arrival time, waiting time, start and end of service are updated.  $MaxShift$  values are also updated for the locations from the starting point to the ending point of the route. The process is repeated for as long as  $t_{max}$  is not exceeded and the set  $V_{R_2-R_1}$  is not empty. Similarly,  $PR(R_2, R_1)$  is performed by inserting vertices from  $V_{R_1-R_2}$  into  $R_2$ . In result two new routes are created. If the fitness values of the new routes are higher than the fitness value of  $R_1$  and  $R_2$ , they replace them.

### 3.4 Mutation

In mutation phase a random route is selected from all routes. Two types of mutation are used by us a gene insertion or gene removal (the probability of each is 0.5). The mutation process is repeated on the selected route  $N_m$  times, where  $N_m$  is the parameter of the algorithm. In *insertion mutation*, all possibilities for inclusion of each new POI not presented in the route are considered. We check whether the shift resulting from the new insertion exceeds the constraints associated with the previously calculated *wait* and *MaxShift* values of the POIs located directly after the newly inserted one. The location  $u$  with the highest value of  $(p_u)^2/TravelTimeIncrease(u)$  is selected for insertion.  $TravelTimeIncrease(u)$  is defined as the increased travel and visit time when POI  $u$  is included. After the insertion POI  $u$  into the route the values of  $arrival_u$ ,  $wait_u$ ,  $start_u$  and  $end_u$  are calculated. For each location after  $u$  the arrival time, waiting time, start and end of service are updated. The *MaxShift* value is updated for each location in the tour.

In the *deletion mutation* we remove a randomly selected POI in order to shorten the travel length. After the gene is removed, all locations after the removed gene are shifted towards the beginning of the route. Furthermore, the locations before and after the removed POI should be updated.

## 4 Computational Experiment

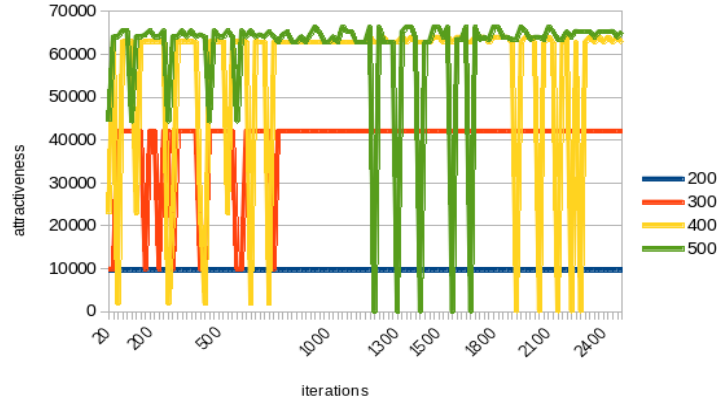
The EA was implemented in C++ and run on an Intel Core i7, 1.73 GHz CPU (turbo boost to 2.93 GHz). The computational experiments were carried out on realistic database 531 POIs and EV charging stations Podlasie region (Poland). Each POI is specified by coordinates, profit (determined on the basis of the votes of Internet users on the given object), time window and visiting time (randomly assigned). Shortest car distance between each pair of POIs and between POIs and battery charging stations were determined on the basis of the queries to Bing Map Service (Routes API).

For comparison results we also tests another version of EA having crossover instead path relinking process. The version with crossover is denoted by EA CR and the version with path relinking by EA PR. In the crossover we selected two routes and in these routes we determine all genes which could be replaced without violating time windows conditions and the limit of route length. Finally a random pair is selected from all similar pairs of genes and this is a point of crossover. The new routes are created by exchange chromosome fragments from both parents from crossover point to the end of route.

Many tests were carried out to establish the evolutionary algorithms parameters and determine convergence and sensitivity. The parameter values which represent the best trade-off between the quality of the solutions and the computational time are as follows: the number of routes in the initial population-  $P_{size}=150$ , the number of groups in the tournament selection-  $k=15$ , the number of routes removed from the group  $t_{size}=10$ , the mutation process is repeated on the selected route  $N_m=15$  times. In tests we take different limit of routes length

( $t_{max}$ ) because of different number of kilometers possible to drive on a single battery charge (this value depends on an EV brand and do not exceed 500 km). We recall that by the attractiveness we mean sum of scores connected with POI's included to the route. The attractiveness of routes balance (see Fig. 1) because of removing POI in deletion mutation. It can be notice that the attractiveness is not higher after 1000 iterations so the maximum number of algorithm generations was set to 1000. Additionally after each 100 generations the current population is checked and the algorithm is stopped if no improvements in attractiveness of route have been found.

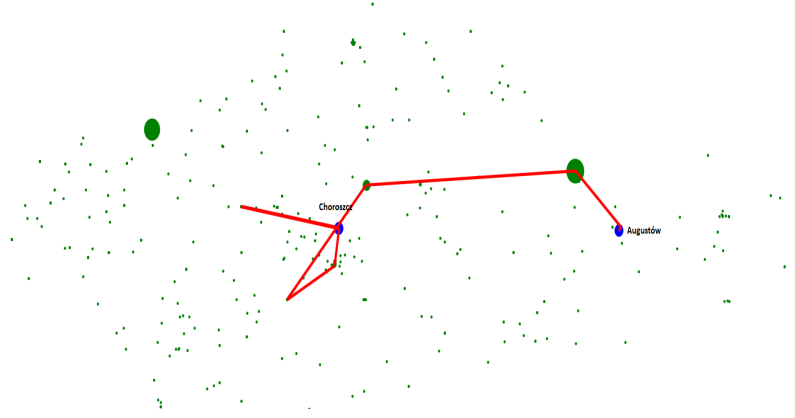
The comparison of the results according to the EA CR (the version with



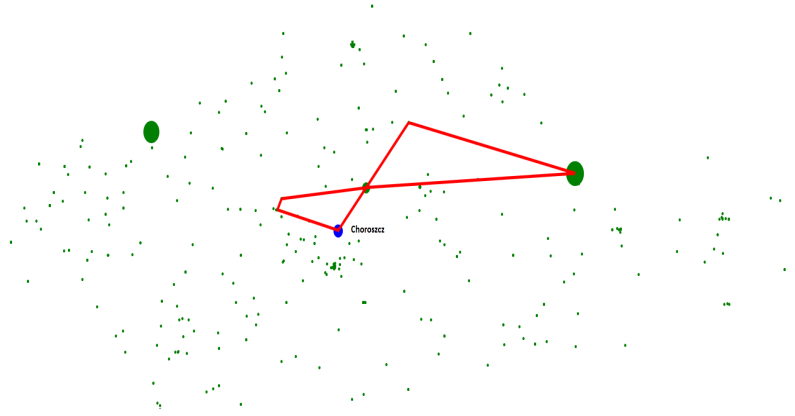
**Fig. 1.** Convergence of the GA PR for  $t_{max}$  = 200-500 value

crossover) as well as EA PR (the version with the path relinking) are outlined in Tab. 1 and Tab. 2. In tables the length is given in kilometers, execution time of the algorithm in seconds. The algorithm is run 15 times and in tables we place the best results from all runs. The execution time is calculated as a sum from 15 runs of algorithm. Results indicate that both algorithms give results in the comparable execution time (about 0.2 s). In the case when the starting and ending points are different (see Tab. 1) both versions give the same results in the length of route and attractiveness. The exception is  $t_{max}$ =300 where EA PR gives better result in attractiveness than EA CR about 2%. When the starting and ending point is the same for  $t_{max}$ =400, 500 (see Tab. 2) the length of routes is higher in case of EA PR than EA CR. However, in case of  $t_{max}$ =400 for EA CR the attractiveness of a route is higher about 0.7% in comparison to EA PR. Both algorithms for  $t_{max}$ = 200, 300 give the same attractiveness and the length. For  $t_{max}$ =500 results EA PR outperform EA CR about 0.1%. Fig. 1, 2 illustrate generated routes by EA PR marked on Podlasie region map. In figures green

points denotes POIs and their size depend on POI attractiveness. The starting and ending points (battery charging stations) are blue.



**Fig. 2.** EV route example: 1-352-24-370-56-344-441- 442-5-2 (the battery charging stations are different).



**Fig. 3.** EV route example: 1-2-357-442-344-433-381- 431-23-1. (the battery charging station is the same)

**Table 1.** Comparison EA CR and EA PR routes (starting and ending charge stations are different)

$t_{max}$	EA CR			EA PR		
	length	attractiveness	time	length	attractiveness	time
200	198.970	1427	0.164	198.970	1427	0.168
300	287.640	42629	0.154	296.420	43497	0.159
400	393.224	51206	0.177	393.224	51206	0.174
500	499.152	65460	0.221	499.152	65460	0.187

**Table 2.** Comparison EA CR and EA PR routes (the starting and the ending charge station is the same)

$t_{max}$	EA CR			EA PR		
	length	attractiveness	time	length	attractiveness	time
200	198.324	9695	0.165	198.324	9695	0.164
300	285.147	42103	0.166	285.147	42103	0.160
400	382.272	64014	0.171	386.839	63569	0.171
500	488.931	66411	0.193	497.797	66466	0.209

## 5 Conclusions and Further Work

In this paper we describe the problem of generation EV tourist routes which maximizing the attractiveness of a trip and the length do not exceeded the number of kilometers that a car can travel on a single battery charge . The problem is transformed to the graph routing problem named the Orienteering Problem with Time Windows. The evolutionary algorithm is presented by us as the solution of the problem. The tests on the real dates show that two versions of algorithm generate routes with similar attractiveness. The execution time of the algorithm is relatively short to use them in practice.

Modern electric vehicles like BMWi3, Tesla S have navigation system to make driving easy and convenient. For example the range assistant with dynamic range map indicate the maximum distance that the vehicle can achieve in all directions from current location and the route is modified in real time. The navigation system in these cars do not select the set of POIs maximizing tourist satisfaction and do not take into account the trip constraints such as: driving range of a car as well as POIs visiting hours. In this regard the evolutionary algorithm described in this paper could be a module of the software application which suggest a driver the most attractiveness trip and the length of the route do not exceed a battery range.

It easy notice that OPTW model allows describe route which are only one stage (i.e. the route ends when a battery is discharged). In our further research it is our intention to generate multistage route i.e. we want maximize the attractiveness of EV tourist routes which be continue after charging battery. In this case Team Orienteering Problem with Times Windows models such problem [14].

## Acknowledgements

The authors gratefully acknowledge support from the Polish Ministry of Science and Higher Education at the Bialystok University of Technology (grant S/WI/1/2014 and S/WI/2/2013).

## References

1. Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., Margaritis, D.: Electric Vehicle Routing Problem with Industry Constraints: Trends and Insights for Future Research. *Transportation Research Procedia*. **3** 452–459 (2014). doi:10.1016/j.trpro.2014.10.026
2. Campos, V., Marti, R., Sanchez-Oro, J., Duarte, A.: Grasp with Path Relinking for the Orienteering Problem. *Journal of the Operational Research Society*. (2013). doi:10.1057/jors.2013.156
3. Dekker, R., Bloemhof, J., Mallidis, I.: Operations Research for green logistics An overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*. **219(3)** 671–679 (2012). doi: 10.1016/j.ejor.2011.11.010
4. EV rentals in Alps. Last access: February 16, 2017. [www.greencarcongress.com/2010/08/think-and-migros-teamup-with-swiss-ecotourism-group-to-offer-ev-rentals-in-alps.html](http://www.greencarcongress.com/2010/08/think-and-migros-teamup-with-swiss-ecotourism-group-to-offer-ev-rentals-in-alps.html)
5. EV Tourism in US. Last access: February 16, 2017. <http://fresh-energy.org/wp/wp-content/uploads/2014/08/White-Paper-EV-Tourism-2014.pdf>
6. EV trip planner. February 16, 2017. <https://www.evtriplanner.com>
7. Glover, F.: A Template For Scatter Search And Path Relinking, *Lecture Notes in Computer Science*. **1363** 13–54 (1997). doi:10.1007/BFb0026589
8. Huang, Y. H., Ting, C. K.: Genetic algorithm with Path Relinking for the Multi-Vehicle Selective Pickup and Delivery problem, *Evolutionary Computation (CEC), IEEE Congress*. 1818–1825 (2011). doi:10.1109/CEC.2011.5949836
9. Jia, S., Hu, Z.H.: Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem. *Computers and Operations Research*. **47** 11–26 (2014). doi:10.1016/j.cor.2014.01.010
10. Kantor, M., Rosenwein, M.: The Orienteering Problem with Time Windows. *Journal of the Operational Research Society*. **43** 629–635 (1992). doi:10.2307/2583018
11. Karbowska-Chilinska, J., Zabielski, P.: A Genetic Algorithm Solving Orienteering Problem with Time Windows. *Advances in Systems Science. Advances in Intelligent Systems and Computing*, Springer. 240 609–619 (2014). doi:10.1007/978-3-319-01857-7\_59
12. Karbowska-Chilinska, J., Zabielski, P.: Genetic Algorithm with Path Relinking for the Orienteering Problem with Time Windows. *Fundamenta Informaticae*. **135(4)** 419–431 (2014). doi:10.3233/FI-2014-1132
13. Lin, C., Choy, K.L., G.T.S. Ho., Chung S.H., Lam H.Y.: Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications*. **41(4)** 1118–1138 (2014). doi:10.1016/j.eswa.2013.07.107
14. Labadi, N., Mansini, R., Melechovsky, J., Calvo, R. W.: The Team Orienteering Problem with Time Windows: An LP-based Granular Variable Neighborhood Search. *European Journal of Operational Research*. **220(1)** 15–27 (2012). doi:10.1016/j.ejor.2012.01.030

15. Schneider, M., Stenger, A., Goeke, D.: The Electric Vehicle Routing Problem with Time Windows and Recharging Stations. *Transportation Science*. **48(4)** 500–520 (2014). doi:10.1287/trsc.2013.0490
16. Montemanni, R., Gambardella, L.M.: Ant colony system for team orienteering problems with time windows. *Foundations of Computing and Decision Sciences*. **34** (2009)
17. Ostrowski, K., Karbowska-Chilinska, J., Koszelew, J., Zabielski, P.: Evolution-inspired local improvement algorithm solving orienteering problem. *Annals of Operational Research*. 1–25 (2016). doi:10.1007/s10479-016-2278-1
18. Perez, M.P., Rodriguez, F.A., Moreno-Vega, J.M.: A hybrid VNS-path relinking for the p-hub median problem. *Journal Management Mathematics*. **18(2)** 157–171 (2007). doi:10.1093/imaman/dpm013
19. Tang, H., Miller-Hooks, E.: A tabu search heuristic for the team orienteering problem. *Computers & Operational Research*. **32(6)**, 1379–1407 (2005). doi:10.1016/j.cor.2003.11.008
20. Tsiligirides, T.: Heuristic methods applied to orienteering. *Journal of the Operational Research Society*. **35(9)** 797–809 (1984). doi:10.1057/jors.1984.162
21. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. *Computers O.R.* **36** 3281–3290 (2009). doi:10.1016/j.cor.2009.03.008
22. Vansteenwegen, P., Souffriau, W., Van Oudheusden, D.: The Orienteering Problem: A survey, *European Journal of Operational Research*. **209(1)** 1–10 (2011). doi:10.1016/j.ejor.2010.03.045
23. Wen, M., Linde, E., Ropke, S., Mirchandani, P., Larsen, A.: An adaptive large neighborhood search heuristic for the Electric Vehicle Scheduling Problem, *Computers & Operations Research*. **76** 73–83 (2016). doi:10.1016/j.cor.2016.06.013
24. Zabielski, P., Karbowska-Chilinska, J., Koszelew, J., Ostrowski, K.: A genetic algorithm with grouping selection and searching operators for the orienteering problem *Lecture Notes in Computer Science*. **9012** 31–40 (2015). doi: 10.1007/978-3-319-15705-4\_4