THE DESIGN OF CHARGING INFRASTRUCTURE FOR ELECTRIC VEHICLES AND ITS PROPERTIES

Peter Czimmermann
University of Žilina, Žilina, Slovakia, peter.czimmermann@fri.uniza.sk
Michal Koháni
University of Žilina, Žilina, Slovakia, Michal.Kohani@fri.uniza.sk
Ľuboš Buzna
University of Žilina, Žilina, Slovakia, Lubos.Buzna@fri.uniza.sk

Abstract: We study methods to design a private charging infrastructure for a fleet of electric vehicles, for example taxicabs, small vans (used in the city logistics) or shared vehicles. In this contribution, we continue our previous work, where we suggested an MIP model of this location-scheduling optimization problem. We introduce an IP model and simplify it using flows in networks. We suggest an algorithm that finds a feasible solution. We also use the flow version to answer the question from our previous work: What is the cause of the gap in the number of feasible vehicles obtained from the optimization problem and evaluation procedure?

Keywords: electric vehicles, charging infrastructure, flows in networks

1 INTRODUCTION

In [2], we suggested and solved an MIP model for the location-scheduling optimization problem to design private charging infrastructure for electric vehicles. In the extended version [3] of the above mentioned paper, we modified this model and added a procedure which simulates the vehicle charging in specific cases, when drivers operate with incomplete information. The tests have been carried out using real data based on activities of taxicabs in Stockholm. We found out that there is a large gap in the number of feasible vehicles obtained from the solution of the mathematical model and the number of feasible vehicles that form an output of the evaluation procedure. The above mentioned model from the work [3] is in Figure 1.

Here, I is the set of candidate locations, where it is possible to place the charging infrastructure. T is the set of non-overlapping time intervals. The fleet is represented by the set of vehicles C. Vehicles have battery with capacity β (measured in kilometres) and s is the charging speed. R_c is the ordered sequence of parking events of vehicle $c \in C$ and N_{cr} is the list of intervals that overlap with the parking event $r \in R_c$. The fraction of the interval $t \in T$, when the vehicle $c \in C$ is parking, we denote by $a_{ct} \in \langle 0, 1\rangle$. $B_{itc} = 1$ when the vehicle $c \in C$

Model formulation

$$\text{Minimize} \sum_{i \in I} s_i \tag{1}$$

$$\text{subject to} \sum_{c \in C} B_{itc} x_{ct} \leq s_i \qquad \text{for } i \in I, t \in T \tag{2}$$

$$d_{c0} \leq \alpha\beta \tag{3}$$

$$d_{cr} + \sum_{t \in N_{c,r}} a_{ct} x_{ct} s \leq \beta \qquad \text{for } c \in C, r \in R_c \cup \{r_c\} \tag{4}$$

$$d_{cr} \leq d_{c,r-1} - u_{cr} + \sum_{t \in N_{c,r-1}} a_{ct} x_{ct} s \qquad \text{for } c \in C, r \in R_c \cup \{r_c\} \tag{5}$$

Figure 1: Mathematical formulation of the location-scheduling optimization problem.

1542 vehicles				
ρ_{max}	M_{min}	Model	Strategy 1	Strategy 2
100	800	609	401	363
	150	1186	652	536
	100	1287	689	586
500	800	1102	592	589
	150	1442	653	603
	100	1475	709	695
1000	800	1347	739	692
	150	1499	785	701
	100	1510	793	706

Table 1: Numbers of feasible vehicles, obtained from the mathematical model and the charging simulation procedure.

parks at location $i \in I$ during the time interval $t \in T$, and $B_{itc} = 0$ otherwise. Vehicle $c \in C$ drives u_{cr} kilometres while driving from the parking event $r-1$ to the parking event r. We work with the following variables: $s_i \in Z_0^+$ represents the number of charging points placed at the location $i \in I$; $x_{ct} \in \{0,1\}$, where $x_{ct} = 1$ when vehicle $c \in C$ is charged during the time interval $t \in T$ and $x_{ct} = 0$ otherwise; and $d_{cr} \geq 0$ is the distance that the vehicle $c \in C$ is able to drive at the beginning of the parking event $r \in R_c$. In the objective function (1), we minimize the number of located charging points. Constraints (2) ensure that we cannot simultaneously use more charging points in each time interval than is the number of all existing charging points. Constraints (4) ensure that battery capacity is not exceeded and constraints (5) ensure the contiguity of charging and discharging of batteries.

In the charging procedure, we process events in ascending order with respect to their time of occurence. If we have a parking event, then we assign to the vehicle associated with this event a specified charging point according to the chosen strategy. Strategy 1 assigns to each vehicle a free charging point that allows charging to the maximum capacity. Strategy 2 assigns to each vehicle the first available charging point that is free.

The results of the comparison for one selected week can be seen in Table 1.

Where ρ_{max} is the maximum acceptable distance of a vehicle from a charging station, M_{min} is the least number of parking events in a location to be acceptable as a candidate for a charging station. In the remaining three columns, we have the numbers of feasible vehicles, which we obtain as results of optimisation (Model) and the charging evaluation procedure (Strategy 1, Strategy 2). More results and detailed description can be found in [3].

2 DISCRETE MODEL

In this section, we present a discrete version of Model 1. For technical reasons, we omit the constraints (3) and values a_{ct}. Variables d_{cr} should be from the set Z_0^+ of non-negative integers. The model is in Figure 2 and the example can be seen in Figure 3, where $t_1, \ldots, t_7$ are time intervals, i_1, i_2, i_3 are candidate locations and c_1, c_2 are vehicles.

3 SIMPLIFICATION

The discrete model from the previous section is a basis for the reduction that is presented here. We need to state several constraints to obtain a model that will help us to answer the question about the gap in the number of feasible vehicles in the columns in Table 1. The ordered set

Model 2 - discrete model formulation

$$\text{Minimize} \sum_{i \in I} s_i \tag{6}$$

$$\text{subject to} \sum_{c \in C} B_{itc} x_{ct} \le s_i \qquad \text{for } i \in I, t \in T \tag{7}$$

$$d_{cr} + \sum_{t \in N_{c,r}} x_{ct} s \le \beta \qquad \text{for } c \in C, r \in R_c \cup \{r_c\} \tag{8}$$

$$d_{cr} \le d_{c,r-1} - u_{cr} + \sum_{t \in N_{c,r-1}} x_{ct} s \qquad \text{for } c \in C, r \in R_c \cup \{r_c\} \tag{9}$$

Figure 2: Discrete version of the location-scheduling optimization problem.

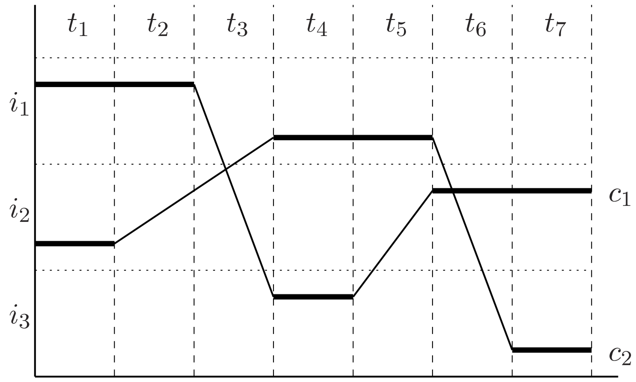

Figure 3: Diagram illustraiting parking events and movements of vehicles.

of all time intervals $T = \{1, 2, \ldots, n\}$ is partitioned into subsets T_{odd} of all time intervals on odd positions and T_{even} of all time intervals on even positions. We suppose that each parking event and journey of the vehicle takes one time interval. The parking events are in even time intervals and journeys are in odd time intervals. We also suppose that at most one charging point can be placed in each candidate location. These restriction can be expressed in Model 2 as follows: $s_i \in \{0, 1\}$, $s = 1$, $u_{c,r} \in \{0, 1\}$, $N_{c,r} = \{2r\}$, $B_{i,t,c}$ and $x_{c,t}$ can be equal to one only in cases $t \in T_{even}$. We call this simplification: Model 3, and we show that it is equivalent to a specific network flow model (explained below).

We construct a network G which represents the situation described above. The set of vertices of G is the union of the sets
$V_1 = \{v_{it} : i \in C, t \in T\}$, where v_{it} represents the vehicle i in time t,
$V_2 = \{u_{it} : i \in I, t \in T_{even}\} \cup \{w_{it} : i \in I, t \in T_{even}\}$, where u_{it} and w_{it} represent the candidate location i in time t,
$V_3 = \{s, z, w\}$, where s is the source, z is the sink and w an additional vertex of the network.

The set of edges is the union of the sets
$E_1 = \{(v_{i,t-1}, v_{i,t}) : i \in C, t \in T - \{1\}\} \cup \{(v_{i,n}, z) : i \in C\}$,
$E_2 = \{(u_{i,t-2}, u_{i,t}) : i \in I, t \in T_{even} - \{2\}\}$,
$E_3 = \{(u_{i,t}, w_{i,t}) : i \in I, t \in T_{even}\}$,
$E_4 = \{(w_{i,t}, v_{j,t}) : i \in I, t \in T_{even}, j \in C, B_{i,t,j} = 1\}$,
$E_5 = \{(v_{i,t}, w) : i \in C, t \in T_{odd}\}$,
$E_6 = \{(s, v_{i,1}) : i \in C\}$,
$E_7 = \{(s, u_{i,2}) : i \in I\}$,

E_1	$\langle 0, \beta \rangle$	E_6	$\langle \beta, \beta \rangle$
E_2	$\langle 0, \|T_{even}\| \rangle$	E_7	$\langle 0, \|T_{even}\| \rangle$
E_3	$\langle 0, 1 \rangle$	E_8	$\langle 0, \|T_{even}\| \rangle$
E_4	$\langle 0, 1 \rangle$	E_9	$\langle 0, \infty)$
E_5	$\langle 1, 1 \rangle$		

Table 2: Lower and upper bounds for the flows on edges.

$E_8 = \{(u_{i,m}, z) : i \in I, m = \max(T_{even})\}$,
$E_9 = \{(w, z)\}$.
Every edge has assigned to it a lower and upper bound for the flow, which are represented by the interval $\langle l, u \rangle$. These bounds are given in Table 2.

The optimal solution of Model 3 is equivalent to the feasible flow in G with the minimum number of non-zero flows on the edges of the set E_7. If the flow on the edge $(s, u_{i,2}) \in E_7$ is non-zero, then we place a charging point in location i. Problems of finding a feasible flow in the network with lower bounds on edges and its minimisation are solvable in polynomial time [1].

We can use the following algorithm to obtain the solution with fewer edges from E_7 with non-zero flow.
Description of the algorithm. We construct a network G' (it is subnetwork of G):
Its vertex set is $V_1' \cup V_2' \cup V_3'$, where
$V_1' = \{v_{i,t} : i \in C, t \in T - \{1\}\} \cup \{z\}$,
$V_2' = \{u_{i,t} : i \in I, t \in T_{even}\} \cup \{w_{i,t} : i \in I, t \in T_{even}\}$.
The edge set is the union of the sets:
$E_1' = \{(v_{i,t-1}, v_{i,t}) : i \in C, t \in T - \{1, 2\}\} \cup \{(v_{i,n}, z) : i \in C\}$,
$E_2' = \{(u_{i,t-2}, u_{i,t}) : i \in I, t \in T_{even} - \{2\}\}$,
$E_3' = \{(u_{i,t}, w_{i,t}) : i \in I, t \in T_{even}\}$,
$E_4' = \{(w_{i,t}, v_{j,t}) : i \in I, t \in T_{even}, j \in C, B_{i,t,j} = 1\}$,
$E_5' = \{(u_{i,m}, z) : i \in I, m = \max(T_{even})\}$.
Lower and upper bounds of the edges are taken from G.

Algorithm.
Input is the network G with feasible flow x and subnetwork G'.
For each pair of vertices $u_{i,2}, u_{j,2} \in V_2$ such that $0 < x(s, u_{i,2}) \leq x(s, u_{j,2}) < |T_{even}|$ **do**:
$\longrightarrow$ **while** there is an augmenting path $P(j, i)$ with reserve r from $u_{j,2}$ to $u_{i,2}$ in G' **do**:
$\longrightarrow$ $\longrightarrow$ add the edges $(s, u_{j,2})$ and $(s, u_{i,2})$ to $P(j, i)$ to form a (non-oriented) cycle
$\longrightarrow$ $\longrightarrow$ $C = (s, u_{j,2}, \ldots, u_{i,2}, s)$,
$\longrightarrow$ $\longrightarrow$ change the flow x in C as follows:
$\longrightarrow$ $\longrightarrow$ **if** (u, v) is the forward edge in C, **then** $x(u, v) = x(u, v) + r$,
$\longrightarrow$ $\longrightarrow$ **if** (u, v) is the reverse edge in C, **then** $x(u, v) = x(u, v) - r$,
$\longrightarrow$ process another pair of vertices.

A polynomial algorithm for finding an augmenting path in network is presented in [4].
This approach does not guarantee that given solution is optimal. We aim to test the algorithm and study its properties in our future works.

Example 1.
Situation with two vehicles, two candidate locations and five time intervals can be seen in Figures 4 and 5.

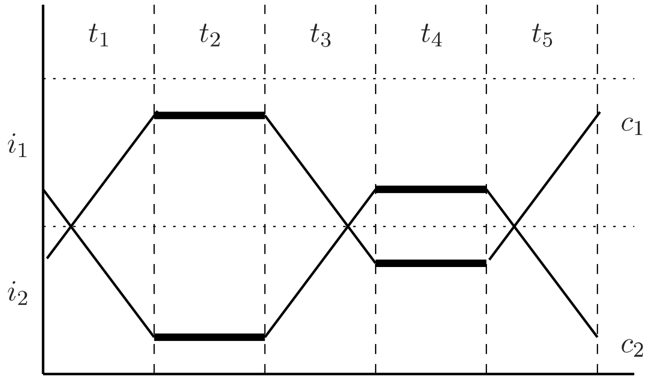

Figure 4: Diagram from Example 1.

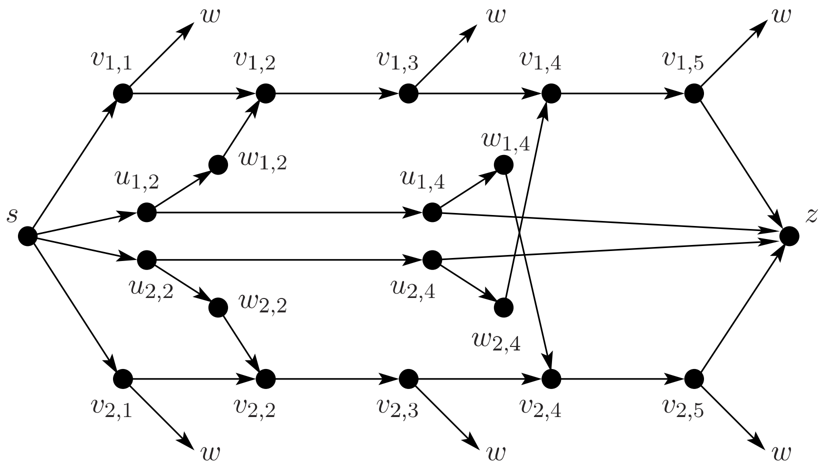

Figure 5: Network from Example 1, $|T_{even}| = 2$.

Question: What is the cause of the gap in the number of feasible vehicles obtained from the optimization problem and simulation procedure?
We suppose that the main reason is in the available information. When we use an optimization algorithm, we work with the whole network G. However, in the simulation procedure, at time $t \in T_{even}$, we only have information about edges from vertices $u_{i,t}$. These constraints do not allow us to find an optimal solution by the evaluation procedure. The open questions are: How do we design the charging infrastructure to be more succesful especially in cases where the constrained level of information is given? How can we spread the available information (from O-D matrices, probability models) to obtain a lower number of unfeasible vehicles?

4 CONCLUSIONS

The main idea of the work is the representation of the simpler version of Model 2 that works with flows in networks. In this approach, we obtain the optimal solution when we find the feasible flow with the minimum number of non-zero flows on the input edges.

We suppose that this approach will allow us to study the role of the available information in the charging process. For example, with this model, we are able to explain the gap between the results of optimization and the simulation procedure, which can be seen in Table 1. In optimization, we have all the information about the network. In the eveluation procedure, we have only the information on neighbourhoods of vertices that have actually been processed.

Our plan is to use this flow model to evaluate the dependency of the level of information about available charging points and suggest an appropriate system of assigning the jobs (journeys) and free charging points for vehicles.

Acknowledgements

This work was supported by the research grants VEGA 1/0463/16 "Economically efficient charging infrastructure deployment for electric vehicles in smart cities and communities", APVV-15-0179 "Reliability of emergency systems on infrastructure with uncertain functionality of critical elements", and it was facilitated by the FP 7 project ERAdiate [621386] "Enhancing Research and innovation dimensions of the University of Zilina in Intelligent Transport Systems".

References

[1] Dantzig, G.B., Fulkerson, D.R., (1954). Minimizing the number of tankers to meet a fixed schedule. Naval Res. Logist. Quart., 1, pp. 217–222

[2] Koháni, M., Czimmermann, P., Váňa, M., Cebecauer, M. and Buzna, Ľ. (2017). Designing Charging Infrastructure for a Fleet of Electric Vehicles Operating in Large Urban Areas. In: Proceedings of the 6th International Conference on Operations Research and Enterprise Systems - Volume 1:ICORES, pp. 360–368.

[3] Koháni, M., Czimmermann, P., Váňa, M., Cebecauer, M. and Buzna, Ľ. (2017). Location-scheduling optimization problem to design private charging infrastructure for electric vehicles. Operations Research and Enterprise Systems, ICORES 2017, Revised Selected Papers. Communications in Computer and Information Science, Springer. (Accepted for publication.)

[4] Nemhauser, G.L., Wolsey, L.A., (1988). Integer and Combinatorial Optimization. Wiley-Interscience series in discrete mathematics and optimization. ISBN 0-471-82819-X, pp. 65