A multi-stage stochastic integer programming approach for locating electric vehicles charging stations under demand uncertainty

Ahmed Kadri¹, Mouna Kchaou-Boujelben¹, Céline Gicquel²

¹ United Arab Emirates University, Al Ain, United Arab Emirates
² Université Paris Saclay, LRI, Université Paris Sud, Orsay, F-91405, France
celine.gicquel@lri.fr

Mots-clés : Electric vehicles, facility location, stochastic programming

1 Problem description and modeling

Electric vehicles (EVs) are one of the promising solutions to face environmental and energy concerns. One of the major barriers towards the large scale adoption of EVs is the lack of charging stations to recharge vehicles during long-distance trips, i.e. during trips whose length exceeds the vehicle range. A key element in making EVs more attractive for long-distance trips is thus to deploy infrastructures consisting of fast-charging stations where drivers can recharge their battery within less than 30 minutes. Nevertheless, building such stations requires large capital expenditures. Their locations should thus be carefully selected so as to maximize the recharging demand satisfaction within the available limited investment budget. This leads to the formulation of a facility location problem.

We model the existing road network of the region under study as a directed graph and seek to identify the optimal location of a predefined number of charging stations in order to maximize the satisfaction of the EV recharging demand. The corresponding max coverage problem displays two specific features:

— Most often, drivers do no carry out a special purpose round trip from their home or workplace to the station in order to recharge the battery of their vehicles but rather recharge it while on their way to another destination. This means that the demand to be satisfied should not be modeled as a set of fixed points in the network but rather as a set of origin-destination flows. A flow will be considered as covered if a vehicle can drive a full round trip from its origin to its destination and back without running out of fuel.

— In view of the limited range of an EV, a driver may have to recharge its battery several times during a trip. Recharging stations should be carefully located on the path followed by the driver so as to make sure that the distance between two consecutive stations on the path does not exceed the vehicle range.

Capar et al. [1] propose a mixed-integer linear programming formulation for this flow-refueling location problem in which the demand is modeled as a set of flows between origin-destination pairs and the limited vehicle range is taken into account. But they consider a single-period deterministic setting. In the present study, we focus on extending their model in order to incorporate two important features: the dynamic multi-period aspect of the problem and the uncertainty on the future recharging demand. Namely, deploying an EV recharging infrastructure is not a one-shot decision but rather a step-by-step roll-out process in which location decisions are made dynamically according to the evolution of the demand and the availability of the investment budget. Moreover, as this deployment plan is likely to span several years, there will be significant uncertainties on the problem input data. Not taking these uncertainties into account means neglecting a critical part of the problem and may lead to suboptimal location decisions.
We propose a multi-stage stochastic integer programming approach in which the uncertainties on the future recharging demand are represented through a scenario tree. More precisely, we consider a multi-stage decision process corresponding to the case where the value of the uncertain parameters unfolds little by little following a discrete-time stochastic process and the station location decisions can be made progressively as more and more information is collected. We thus assume that the station opening decisions relative to period $t$ do not have to be made at the beginning of the planning horizon, but rather at the beginning of period $t$ after the charging demand for periods $1...t$ becomes known. Hence, station opening decisions can be adjusted according to the past and current realizations of the charging demand.

2 Solution approaches and numerical results

To solve the resulting large-size mixed-integer linear program, we develop two solution approaches:

— The first one is an exact solution method based on a Benders decomposition. This is achieved by extending the approach proposed by Arslan and Karasan [2] for the single-period deterministic problem to our multi-period stochastic problem. In the proposed algorithm, the master problem decides upon the station deployment strategy over the whole scenario tree while each sub-problem focuses on evaluating the resulting coverage for each trip at each tree node.

— In view of the numerical difficulty of solving instances of the problem for long planning horizons, we also propose a heuristic approach based on a genetic algorithm. To adapt the general algorithm to our problem, we focused on devising cross-over and mutation operators ensuring that the obtained solutions are feasible and on developing a computationally efficient fitness function evaluation algorithm.

Our numerical results carried out on 120 randomly generated medium-size instances are displayed in Table 1. They show that the two methods perform well as compared to the mathematical programming solver CPLEX, both in terms of solution quality and computation time. Namely, the Benders decomposition algorithm is capable of providing more optimal and feasible solutions than CPLEX, and this within an average computation time decreased by around 40%. Moreover, the genetic algorithm consistently provides near-optimal solutions within an average computation decreased by around 80%.

<table>
<thead>
<tr>
<th></th>
<th>CPLEX solver</th>
<th>Benders decomposition</th>
<th>Genetic algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nb of optimal solutions</td>
<td>68</td>
<td>90</td>
<td>—</td>
</tr>
<tr>
<td>Nb of feasible solutions</td>
<td>78</td>
<td>100</td>
<td>120</td>
</tr>
<tr>
<td>Average optimality gap</td>
<td>—</td>
<td>—</td>
<td>4.7%</td>
</tr>
<tr>
<td>Average computation time</td>
<td>1462s</td>
<td>843s</td>
<td>243s</td>
</tr>
</tbody>
</table>

TAB. 1 – Aggregated results for 120 randomly generated instances

Références
