Efficient solution approaches for locating electric vehicle fast charging stations under driving range uncertainty

Mouna Kchaou-Boujelben, Céline Gicquel

Efficient solution approaches for locating electric vehicle fast charging stations under driving range uncertainty

Mouna Kchaou-Boujelben[1] and Céline Gicquel[2]
[1] United Arab Emirates University, Al Ain, United Arab Emirates
[2] Université Paris Saclay, Orsay, France

Abstract

We seek to determine the best locations for electric vehicle fast charging stations under driving range uncertainty. Two stochastic programming based models have been recently proposed to handle the resulting stochastic flow refueling location problem: a first one maximizing the expected flow coverage of the network, a second one based on joint chance constraints. However, significant computational difficulties were encountered while solving large-size instances. We thus propose two efficient solution approaches for this problem. The first one is based on a new location-allocation type model for this problem and results in a MILP formulation, while the second one is a tabu search heuristic. Our numerical experiments show that when using the proposed MILP formulation, the computation time needed to provide guaranteed optimal solutions is significantly reduced as compared to the one needed when using the previously published MILP formulation. Moreover, our results also show that the tabu search method consistently provides good quality solutions within short computation times.

Keywords: Flow refueling location model, electric vehicle charging station network design, stochastic driving range, stochastic programming, mixed-integer linear programming, tabu search

1 Introduction

According to the International Energy Agency (IEA) [11], transport accounts for nearly one quarter of global energy-related CO2 emissions. Reducing CO2 emissions from transport modes will thus play an important role in building a more sustainable future for the next generations. One way to achieve this target is to encourage drivers to use Electric Vehicles (EVs), i.e. vehicles powered by a battery which has to be recharged by plugging the vehicle into an external source of electric power. The EV market has experienced significant growth over the last few years: in 2016, there were more than 2 million EVs on the road globally and more than 500,000 in the United States alone [12]. However, as indicated by a 2017 report from the US department of energy [22], the widespread market adoption of EVs still remains hindered by many factors, among which is the lack of a convenient and ubiquitous network of charging stations. Namely, the range of EVs, i.e. the maximum distance that a fully charged EV can travel before its battery runs empty, is rather limited as compared to the one of vehicles powered by conventional internal combustion engines. As a consequence, an EV driver wishing to travel a long-distance trip has to make multiple stops at public charging stations during his trip in order to recharge the EV battery. These stations should be easily accessible, when and where needed (typically in gasoline stations), and the charging time should be limited to a few minutes. Currently, the available fast charging technology enables a driver to recharge the battery of his vehicle within 20-30 minutes. However, it remains far less widespread than the slow charging technology, mainly because of its high investment costs.

The deployment of an EV charging infrastructure based on the fast charging technology thus requires to carefully select the locations of the charging stations so as to simultaneously satisfy as much charging demand as possible and limit the investment costs. This combinatorial optimization problem, known as the flow refueling location problem (FRLP), was first introduced in [15]. In the FRLP, the demand is modeled as a set of origin-destination trips to be taken by EV drivers. An EV trip is said to be 'covered' or 'refueled' if the charging stations located on the corresponding path allow an EV driver to travel from his point of departure to his destination and back without running out of fuel, i.e. if the distance between each pair of consecutive stations on the path does not exceed the vehicle range. The objective is to select the best locations for EV charging stations in a way to maximize the total flow of EVs covered in the network while complying with the available limited investment budget.

In [15] and most of the extensions of this work, the EV range is assumed to be deterministically known. However, this seems a rather strong assumption as in practice, the actual range might be subject to high variations due among others to the traffic conditions, the outside temperature, the use of in-car heater or air conditioning and the age of the battery. Neglecting range uncertainties when planning an EV charging infrastructure might lead to suboptimal solutions as the actual demand coverage provided by the location decisions might be overestimated. It is thus necessary to explicitly take into account in the location model the stochastic nature of the vehicle range. De Vries and Duijzer [3] proposed two stochastic programming based models to incorporate the driving range variability in the FRLP. However, their numerical experiments showed that the computation time needed to solve the resulting mixed-integer linear programs (MILP) increases vastly with the problem size and might become prohibitively long for some instances. The main objective of the present work is to tackle this difficulty by proposing two efficient solution approaches for this problem: an exact method relying on new MILP formulations and a heuristic one based on a tabu search algorithm.

Our contribution is thus threefold. First, we present a new MILP formulation for the deterministic variant of the FRLP, where the EV range explicitly appears as an input parameter. This formulation is related to the location-allocation models frequently used in the facility location literature. It mainly relies on the idea that, if an EV trip is covered, then each node belonging to the shortest path between the origin and the destination of the trip should be allocated to (i.e. refueled by) a single charging station situated on the path within the driving range of the vehicle. Second, we extend this formulation to consider two stochastic variants of the FRLP with uncertain EV range: the first one maximizing the expected coverage of the charging demand, the second one based on a chance-constrained formulation. Using randomly generated instances as well as real-life instances, we show that the proposed MILP formulations perform better in terms of computation time than the formulations used in [3], for both the deterministic and stochastic problems. Third, since the computation time remains long for some large-size instances, we propose a heuristic solution method based on a tabu search algorithm. Our numerical results show that it consistently provides good quality solutions in significantly reduced computation times.

The paper is organized as follows. In Section 2, we propose an overview of the related literature. Section 3 introduces the deterministic flow refueling location model (FRLM) and presents the new MILP formulation for this problem. Two extensions of the deterministic model taking into account the stochastic driving range are then considered in Section 4: the expected flow refueling location model (EFRLM) and the chance constrained flow refueling location model (CCFRLM). The tabu search heuristic procedure is described in Section 5. Section 6 analyzes the results of the numerical experiments carried out to assess the numerical performance of the proposed exact and heuristic solution approaches. Finally, some conclusions and suggestions for further research are provided

in Section 7.

2 Position in the literature

The application of facility location models to the design of EV charging infrastructure has required specific research development in recent years. A reason for this is that drivers often do not carry out a special-purpose round trip from their home or workplace to the charging station to recharge their vehicles but rather recharge it while on their way to another destination. This is particularly the case for long distance trips. This implies that the EV charging demand should be modeled as a set of origin-destination trips rather than a set of nodes. Such a demand representation was proposed by Hodgson [7] within his flow capturing location model (FCLM). In the FCLM, an origin-destination flow is assumed to be 'captured' if there is at least one station located anywhere on the shortest path between the origin and the destination of the flow. However, the FCLM does not take into account the limited range of EVs and the fact that multiple stations may be needed to recharge a trip. This was the motivation behind the development of the flow refueling location model (FRLM) by Kuby and Lim in [15]. In this model, a flow on a given path is considered as 'covered' if and only if the stations on the path are located such that the distance between each pair of consecutive stations does not exceed the vehicle range.

In the past few years, the problem of locating charging stations for electric vehicles has attracted considerable attention. For instance, efficient solution approaches based on new MILP formulations were proposed in [2] and [20] while heuristic solution techniques for large-scale problems were studied in [5] and [6]. Extensions of the basic problem taking into account a limited charging capacity of the stations (see e.g. [8] and [21]) or the possibility of using paths slightly deviating from the shortest origin-destination path (see e.g. [10], [14], [18] and [25]) were also considered. However, little attention has been given in the literature to the stochastic nature of the EV charging infrastructure planning problem while in practice, there are significant uncertainties on the input parameters of the optimization problem to be solved.

A first parameter subject to uncertainty is the charging demand as it is difficult to accurately forecast the number of EVs on each trip one or several years ahead: see e.g. [9], [13], [19] and [23] for models taking into account a stochastic charging demand. A second source of uncertainty is the driving range, which might be subject to high variations due among others to the traffic conditions, the outside temperature, the use of air conditioning and the age of the battery. With a stochastic driving range, it is not possible to ensure that a trip will be covered for all possible realizations of the random parameter. Hence, the problem modeling involves computing the coverage probability of each trip as the joint probability that each portion of the trip comprised between two opened charging stations will be shorter than the driving range. To the best of our knowledge, only two recent works, [3] and [17], have studied this problem. De Vries and Duijzer [3] assume that, for a given trip, the realization of the driving range will be the same on each portion of the trip and exploit this assumption to ease the computation of the joint probability. They introduce two stochastic programming based models. The first one defines the coverage probability of each trip as a decision variable and seeks to maximize the expected EV flow covered. The second one is a chance constrained model in which the coverage probability of each covered trip is required to stay above a minimum value. In both cases, the problem is reformulated as a MILP and solved with a mathematical programming solver. Lee and Han [17] also seek to maximize the expected EV flow coverage but they assume that, for a given trip, the random variables representing the stochastic driving range realization on each trip portion are totally independent of one another.

This assumption enables them to compute the joint probability as the product of the reachability probability of each trip portion. The problem is then formulated as a mixed-integer non-linear program and solved using a Branch-and-Price approach.

The present work is mostly related to the work of De Vries and Duijzer [3] as we use the same assumption on the driving range realization. The authors of [3] propose a new formulation of the deterministic FRLP in which the vehicle range explicitly appears as an input parameter and exploit it in their two stochastic programming based models. However, their numerical results show that the computation time needed to solve the problem with a mathematical programming solver becomes prohibitively long when the problem size increases. In what follows, we seek to partially remedy to this difficulty and to reduce the computation time by proposing two new efficient solution approaches for the FRLP with a stochastic driving range.

3 Deterministic Flow Refueling Location Problem

3.1 Problem description

We consider a road network $\mathcal{G}(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ denotes a set of nodes and $\mathcal{A}$ denotes a set of arcs linking these nodes. The demand is modeled as a set $\mathcal{Q}$ of predetermined trips to be taken by EV drivers. Each trip q corresponds to traveling from a given origin $O_q \in \mathcal{N}$ to a given destination $D_q \in \mathcal{N}$ and back. We assume that all drivers taking trip q follow the shortest path between O_q and D_q and are not willing to deviate from it to refuel their vehicles. The flow f_q on trip q corresponds to the number of vehicles traveling between O_q and D_q per unit of time.

Similarly to e.g. [2] and [3], we assume that the battery consumption is directly proportional to the distance traveled and that all EVs have the same limited range R (expressed in miles or kilometers). The problem is that the distance traveled during a trip q is likely to exceed the driving range R. In this case, multiple stops at charging stations are required to recharge the EV battery. We assume that charging stations are uncapacitated and accessible for traffic from both directions. In order for an EV to travel from O_q to D_q and back without running out of charge, there should be an adequate number of stations at carefully selected nodes on the shortest path between O_q and D_q. When the distance between each pair of consecutive stations on this path does not exceed the range R, trip q is said to be covered. But as soon as there is a pair of consecutive stations separated by a distance greater than R, trip q is considered not covered. The optimization problem thus consists in identifying the best locations to build a predefined number p of charging stations in the network so as to maximize the total flow of EVs covered.

We seek to enable EV drivers to carry round trips from their origin to their destination. Therefore, a driver taking trip q cyclically travels from O_q to D_q to O_q, etc... alternatively crossing each arc belonging to the shortest path in the outward and return direction. To model the problem, De Vries and Duijzer [3] introduced the concept of cycle segment. As this concept will be used in the MILP formulations of the problem presented in the next two subsections, we discuss it in more detail in what follows.

A cycle segment $[k, l]$ is defined as the sequence of consecutive arcs to be crossed (in the outward and/or return direction) to travel on the shortest path between node k and node l. Note that there is an exception for the cycle segment $[O_q, l]$ that might be understood in two different ways: either the driver travels in the outward direction from O_q to l or he carries out a round trip $l \longrightarrow O_q \longrightarrow l$. Similarly, $[k, D_q]$ might correspond to the situation where the driver travels in the outward direction from k to D_q or where he carries out the round trip $k \longrightarrow D_q \longrightarrow k$. In order

to distinguish between these two cases, for each trip q, nodes O_q and D_q are duplicated into nodes O'_q and D'_q and two arcs of length 0, (O'_q, O_q) and (D_q, D'_q) are added to $\mathcal{A}$. Cycle segment $[O_q, l]$ (resp. $[k, D_q]$) represents traveling in the outward direction from O_q to l (resp. from k to D_q) while cycle segment $[O'_q, l]$ (resp. $[k, D'_q]$) represents the round trip to the origin (resp. destination) of the trip.

Let $\mathcal{N}_q$ (resp. $\mathcal{A}_q$) denote the set of nodes (resp. arcs) belonging to the shortest path between nodes O'_q and D'_q. $\mathcal{A}_q^{k,l}$ represents the set of consecutive arcs visited when traveling on the shortest path between k and l during trip q and $d(m,n)$ is the length of arc $(m,n) \in \mathcal{A}$. The length t_q^{kl} of a cycle segment $[k,l]$ on trip q is defined as follows:

- For $k = O'_q$ and $l \in \mathcal{N}_q \setminus \{D'_q\}$, $t_q^{kl} = 2\sum_{(m,n)\in\mathcal{A}_q^{k,l}} d(m,n)$.
- For $k \in \mathcal{N}_q \setminus \{O'_q\}$ and $l = D'_q$, $t_q^{kl} = 2\sum_{(m,n)\in\mathcal{A}_q^{k,l}} d(m,n)$.
- For $k \in \mathcal{N}_q \setminus \{O'_q, D'_q\}$ and $l \in \mathcal{N}_q \setminus \{O'_q, D'_q\}$, $t_q^{kl} = \sum_{(m,n)\in\mathcal{A}_q^{k,l}} d(m,n)$.
- For $k = O'_q$ and $l = D'_q$, $t_q^{kl} = +\infty$.

These notations are summarized in Table 1.

De Vries and Duijzer [3] recently proposed a formulation of the flow refueling location problem (FRLP) relying on the concept of cycle segment. In this formulation, denoted FRLM1 in what follows, each round trip $q \in Q$ from O'_q to D'_q and back is decomposed into a sequence of sub-trips between charging stations where the EV successively stops to get refueled. Each of these sub-trips corresponds to one of the cycle segments defined above. Thus, a feasible sequence of cycle segments or sub-trips, complying with the stations opening decisions, is built for each trip q. If all cycle segments used for trip q have a length smaller than the range R, the trip is considered as covered. Otherwise, trip q is not covered. Note that, contrary to the MILP formulations investigated e.g. in [2], [15] and [20], formulation FRLM1 contains the range R explicitly as a parameter, which is essential in order to be able to extend it to the stochastic range case. De Vries and Duijzer [3] tested formulation FRLM1 on a large set of instances and encountered significant computational difficulties while solving large-size instances of the problem. In order to decrease the computation time needed to obtain guaranteed optimal solutions, we introduce in Subsection 3.2 a new formulation of the problem (denoted FRLM2).

3.2 New formulation for the deterministic flow refueling location problem

Formulation FRLM2 relies on the idea that, if a trip q is covered, then each node $l \in \mathcal{N}_q$ should be reachable (or in an equivalent manner, all arcs $(j,l) \in \mathcal{A}_q$ should be crossable) by the EV after refueling at an opened charging station located before l. Thus, in case trip q is covered, we should assign to each node $l \in \mathcal{N}_q$ a single station $k \in \mathcal{N}_{lq}^b$ in charge of serving it. This station should be within the driving range R of the vehicle to enable the EV driver to refuel his vehicle at node k and reach node l. In case trip q is not covered, no node-station assignment is needed.
Similarly to formulation FRLM1, formulation FRLM2 uses binary location and coverage variables defined by:

- $x_k = 1$ if a station is built at node k, 0 otherwise;
- $y_q = 1$ if the trip q is covered, 0 otherwise.

Parameter	Description
$\mathcal{N}$	Set of nodes of the road network
$\mathcal{A}$	Set of arcs (i,j) of the road network, $i \in \mathcal{N}, j \in \mathcal{N}$
$d(i,j)$	Length of arc $(i,j) \in \mathcal{A}$
R	Range of electric vehicles
p	Number of charging stations to be built
$\mathcal{Q}$	Set of origin-destination trips taken by EV drivers in the network
f_q	Flow of EVs on trip q
O_q	Origin of trip q
O'_q	Duplicated origin of trip q
D_q	Destination of trip q
D'_q	Duplicated destination of trip q
$\mathcal{N}'$	Set of duplicated nodes
$\mathcal{N}_q$	Set of nodes belonging to the shortest path between O'_q and D'_q
$\mathcal{N}^b_{lq}$	Set of nodes belonging to $\mathcal{N}_q$ visited before node l when traveling from O'_q to D'_q
$\mathcal{N}^a_{kq}$	Set of nodes belonging to $\mathcal{N}_q$ visited after node k when traveling from O'_q to D'_q
$\mathcal{A}_q$	Set of arcs belonging to the shortest path between O'_q and D'_q
$\mathcal{A}^{kl}_q$	Set of consecutive arcs visited when traveling on the shortest path between k and l during trip q
t^{kl}_q	Length of a cycle segment $[k,l]$ on trip q

Table 1: FRLM parameters

In formulation FRLM2, we introduce a set of assignment variables w^{kl}_q defined as follows. For each trip q, each node $l \in \mathcal{N}_q \setminus \{O'_q\}$ and each node $k \in \mathcal{N}^b_{lq}$, we define:

- for $k \neq O'_q$: $w^{kl}_q = 1$ if the EV battery is recharged at the station located at node k to enable the driver to travel in the outward direction at least up to node l, i.e. if node l is assigned to the station opened at node k, and 0 otherwise.
- for $k = O'_q$: $w_q^{O'_q,l} = 1$ if the refueling of the sub-trip $l \longrightarrow O'_q \longrightarrow l$ is ensured by a station located at node l or after it on the shortest path corresponding to q and visited by the EV while traveling in the return direction from D'_q to O'_q.

Using these decision variables and the parameters defined in Table 1, FRLM2 is formulated as follows:

$$
\begin{cases}
Max \sum_{q \in \mathcal{Q}} f_q y_q & & (1)\\
s.t. \quad \sum_{k \in \mathcal{N}^b_{lq}} w^{kl}_q = y_q & \forall q \in \mathcal{Q}, l \in \mathcal{N}_q \setminus \{O'_q\} & (2)\\
w^{kl}_q \leq x_k & \forall q \in \mathcal{Q}, k \in \mathcal{N}_q \setminus \{O'_q, D'_q\}, l \in \mathcal{N}^a_{kq} & (3)\\
\sum_{k \in \mathcal{N}^b_{lq}} t^{kl}_q w^{kl}_q \leq R & \forall q \in \mathcal{Q}, l \in \mathcal{N}_q \setminus \{O'_q\} & (4)\\
\sum_{j \in \mathcal{N} \setminus \mathcal{N}'} x_j = p & & (5)\\
x_k \in \{0,1\} & \forall k \in \mathcal{N} & (6)\\
y_q \in \{0,1\} & \forall q \in \mathcal{Q} & (7)\\
w^{kl}_q \in \{0,1\} & \forall q \in \mathcal{Q}, l \in \mathcal{N}_q \setminus \{O'_q\}, k \in \mathcal{N}^b_{lq} & (8)
\end{cases}
$$

The objective function (1) seeks to maximize the total covered flow computed as the weighted sum of covered trips. Constraints (2) can be seen as assignment constraints stating that, if trip q is covered, each node $l \in \mathcal{N}_q \setminus \{O'_q\}$ on this path should be assigned to a single node $k \in \mathcal{N}^b_{lq}$. Note how constraints (2) set to 0 all variables w^{kl}_q corresponding to trip q in case trip q is not covered. Constraints (3) define the relationship between variables w and variables x: if there is no station opened at node k, no EV can be refueled at node k and all variables w^{kl}_q corresponding to node k should be set to 0. Constraints (4) are the range constraints. They ensure that, if trip q is covered, then each node l on this path is assigned to a station located within a distance smaller than R. Note that, in case trip q is not covered, all w^{kl}_q variables corresponding to trip q are set to 0 and constraints (4) become inactive. Constraint (5) sets the number of charging stations that must be opened to a predetermined number p. Constraints (6)-(8) define the problem variables.

We would like to point out that even if formulation FRLM2 seems to focus only on enabling the driver to reach each node l of a covered trip q while traveling in the outward direction, they also ensure that the round trip between O'_q and D'_q will be feasible thanks to the use in constraints (4) of the cycle segment length t^{kl}_q defined in Subsection 3.1. In order to clarify this point, let us consider a feasible solution of formulation FRLM2 in which a given trip q is covered. We denote $[\pi(1), ..., \pi(s), \pi(s+1), ..., \pi(S)]$ the index of the opened stations successively visited by an EV driver following the shortest path corresponding to q in the outward direction. We now show that if this sequence enables the driver to carry out the trip in the outward direction, it also enables him to travel the whole round trip. Namely:

- Let consider first a node l situated between two stations $\pi(s)$ and $\pi(s+1)$ on path q. Node l will be served by station $\pi(s)$ in the outward direction and by station $\pi(s+1)$ in the return direction. Moreover, since the considered solution is feasible, we have: $w_q^{\pi(s),\pi(s+1)} = 1$ and $t_q^{\pi(s),\pi(s+1)} \leq R$, which means that both stations are located at a distance smaller than R from l. Hence, node l is reachable in both directions.

- Let now focus on a node l located between the origin O'_q and station $\pi(1)$. We have $w_q^{O'_q,\pi(1)} = 1$ with $t_q^{O'_q,\pi(1)} \leq R$. Thanks to the definition of the cycle segment length $t_q^{O'_q,\pi(1)}$, this implies that the round trip $\pi(1) \longrightarrow O'_q \longrightarrow \pi(1)$ is feasible with respect to the driving range R. As the round trip $l \longrightarrow O'_q \longrightarrow l$ is shorter than the round trip $\pi(1) \longrightarrow O'_q \longrightarrow \pi(1)$, node l is reachable in both directions.

- Finally, let consider a node l located between station $\pi(S)$ and the destination D_q. We have $w_q^{\pi(S),D_q} = 1$ with $t_q^{\pi(S),D_q} \leq R$. Thanks to the definition of the cycle segment length $t_q^{\pi(S),D_q}$, this implies that the round trip $\pi(S) \longrightarrow D'_q \longrightarrow \pi(S)$ is feasible with respect to the driving range R. As the round trip $l \longrightarrow D'_q \longrightarrow l$ is shorter than the round trip $\pi(S) \longrightarrow D'_q \longrightarrow \pi(S)$, node l is reachable in both directions.

4 Flow Refueling Location Problem under a stochastic driving range

In practice, the driving range R of an EV is subject to many uncertainties due to e.g. the traffic conditions, the weather or the driving style. Taking the stochastic nature of the driving range into account has a significant impact on the problem modeling. Indeed, in the deterministic models discussed in Section 3, depending on the station location decisions, a trip is either covered or not.

In contrast, when the driving range is stochastic, coverage becomes a matter of chance rather than of binary observation [3]. Namely, in this case, it might not be possible anymore to ensure that a trip will be refueled for all possible realizations of the range, i.e. that a trip will be covered with a 100% probability, even if we open many stations on the corresponding path. In the problem modeling, we will thus consider the coverage probability of a trip which is defined as the probability that an EV driver can travel a round trip from his origin to his destination without running out of fuel. De Vries and Duijzer ([3]) proposed two stochastic programming based models for this problem:

- the expected flow refueling location model where the coverage probability of each trip is a decision variable taking any value between 0 and 1 and the objective is to maximize the expected flow covered.
- the chance constrained flow refueling location model where a trip is considered as covered if its coverage probability remains above a predefined minimum value and the objective is to maximize the total flow covered.

In both cases, De Vries and Duijzer [3] used the following assumptions:

- For a given realization of the random conditions ω, the value of the EV range $R(\omega)$ is the same in all the network.
- $R(\omega)$ is randomly distributed with cumulative density function $G : \Re \rightarrow [0, 1]$

In what follows, we discuss deterministic equivalent reformulations of the stochastic problem. We consider the expected flow refueling location model in Subsection 4.1 and the chance constrained flow refueling location model in Subsection 4.2.

4.1 Expected flow refueling location model (EFRLM)

The expected flow refueling location model (EFRLM) seeks to maximize the total expected flow covered in the network, i.e. the expected number of EV drivers who can complete their trip without running out of fuel. The expected flow covered on a trip q is computed as the product of the flow on trip q (f_q) by the coverage probability of trip q denoted z_q. In the EFRLM, z_q is a decision variable whose value depends on the station location decisions.

De Vries and Duijzer proposed in [3] a formulation (denoted here EFRLM1) for the expected flow refueling location problem, which is based on their formulation FRLM1 used in the deterministic case. In EFRLM1, the binary coverage variables y_q are replaced by continuous variables $z_q \in [0, 1]$ representing the coverage probability of trip q. z_q is the probability that an EV driver can travel a round trip along the path without running out of fuel, i.e. the probability that all successive stations visited by the driver while traveling from O'_q to D'_q and back are located from each other at a distance smaller than the driving range. Hence, z_q can be computed as the joint probability that all cycle segments used by the driver to carry out a round trip have a length smaller than the driving range.

Our new formulation EFRLM2 is based on the formulation FRLM2 proposed for the deterministic problem. Similarly to what is done in formulation EFRLM1, continuous variable $z_q \in [0, 1]$ represents the coverage probability of trip q. Here, z_q is computed as the joint probability that all nodes $l \in \mathcal{N}_q \setminus \{O'_q\}$ are assigned to a station $k \in \mathcal{N}^b_{lq}$ located at a distance t^{kl}_q smaller than the driving range.

We thus have:

$$z_q = P\Big(R(\omega) \geq t_q^{kl} w_q^{kl},\ \forall l \in \mathcal{N}_q \setminus \{O'_q\}, \forall k \in \mathcal{N}_{lq}^b\Big) \tag{9}$$

$$= P\left(R(\omega) \geq \max_{(k,l)} \left\{t_q^{kl} w_q^{kl}\right\}\right) \tag{10}$$

$$= 1 - G\left(\max_{(k,l)} \left\{t_q^{kl} w_q^{kl}\right\}\right) \tag{11}$$

$$= 1 - \max_{(k,l)} \left\{G\big(t_q^{kl} w_q^{kl}\big)\right\} \tag{12}$$

$$= 1 - \max_{(k,l)} \left\{G\left(t_q^{kl}\right) w_q^{kl}\right\} \tag{13}$$

$$= 1 - \max_{(k,l)} \left\{g_q^{kl} w_q^{kl}\right\} \tag{14}$$

Equality (10) makes use of the assumption that the driving range realization $R(\omega)$ is the same in all the network and computes z_q as the probability that the driving range is greater than the length of the longest cycle segment used to travel on trip q. In (11), this probability is expressed using the cumulative density function G. (12) holds because G is a non-decreasing function. Then, we have $G\big(t_q^{kl} w_q^{kl}\big) = G\left(t_q^{kl}\right) w_q^{kl}$ thanks to the fact that w_q^{kl} is a binary variable and $G(0) = 0$, which gives equality (13). Finally, by defining $g_q^{kl} = G\left(t_q^{kl}\right) = P\left(R(\omega) \leq t_q^{kl}\right)$, we obtain (14).

In the MILP formulation, the value of z_q can thus be computed through a set of linear constraints of the form $z_q \leq 1 - g_q^{kl} w_q^{kl}, \forall l \in \mathcal{N}_q \setminus \{O'_q\}, \forall k \in \mathcal{N}_{lq}^b$. Moreover, we have the property that, in any optimal solution of the problem, for each trip q and each node $l \in \mathcal{N}_q \setminus \{O'_q\}$, there will be at most one variable w_q^{kl} equal to 1. This enables us to aggregate these constraints into $z_q \leq 1 - \sum_{k \in \mathcal{N}_{lq}^b} g_q^{kl} w_q^{kl}, \forall l \in \mathcal{N}_q \setminus \{O'_q\}$.

This leads to the following formulation EFRLM2:

$$\begin{cases} Max \sum_{q \in \mathcal{Q}} f_q z_q & & (15) \\ s.t.\ \ z_q \leq \sum_{k \in N_{lq}^b} w_q^{kl} & \forall q \in \mathcal{Q}, l \in N_q \setminus \{O'_q\} & (16) \\ \quad z_q \leq 1 - \sum_{k \in \mathcal{N}_{lq}^b} g_q^{kl} w_q^{kl} & \forall q \in \mathcal{Q}, l \in \mathcal{N}_q \setminus \{O'_q\} & (17) \\ \quad z_q \in [0,1] & \forall q \in \mathcal{Q} & (18) \\ \quad (3), (5), (6), (8) & & \end{cases}$$

Constraints (16) state that if a trip q has a strictly positive probability of coverage ($z_q > 0$), for each node $l \in \mathcal{N}_q \setminus \{O'_q\}$, there must exist a node k visited before l during trip q, where the vehicle can be refueled up to node l. Constraints (17) link the coverage probability variables z_q to the binary variables w_q^{kl}. They state that variable z_q is calculated as the smallest coverage probability over all segments $[k, l]$ where node l is refueled by a station at node k. Constraints (3), (5), (6) and (8) are maintained from the deterministic formulation FRLM2.

4.2 Chance constrained flow refueling location model (CCFRLM)

As explained by De Vries and Duijzer [3], the expected flow refueling location model may have some drawbacks when considering pure (rather than hybrid) electric vehicles. Namely, this model

implicitly assumes that EV drivers will be ready to undertake a trip q as soon as its coverage probability is positive. However, in practice, an EV driver might not be willing to take a trip q if the probability of running out of fuel during the trip is above a maximum acceptable risk level α. De Vries and Duijzer [3] thus introduced the chance constrained flow refueling location model CCFRLM which seeks to maximize the number of drivers for which the probability of completing their trip without running out of fuel is at least $1-\alpha$. De Vries and Duijzer developed formulation CCFRLM1, which is the chance-constrained extension of their deterministic model FRLM1.

We propose a new formulation CCFRLM2, which is a chance-constrained extension of the deterministic model FRLM2 proposed in Subsection 3.2. Similarly to formulation CCFRLM1, in CCFRLM2, a trip q is considered covered if its coverage probability is higher than $1-\alpha$ and uncovered otherwise.

Using inequalities (14) to compute the coverage probability, we have for each trip q:

$$\begin{cases} 1-\max\limits_{k,l}\left\{g_q^{kl}w_q^{kl}\right\} \geq 1-\alpha, & \text{if } y_q=1 \\ 1-\max\limits_{k,l}\left\{g_q^{kl}w_q^{kl}\right\} \geq 0 & \text{if } y_q=0 \end{cases}$$

which can be reformulated through the set of linear constraints $\sum_{k\in\mathcal{N}_{lq}^b} g_q^{kl}w_q^{kl} \leq \alpha, \forall l \in \mathcal{N}_q \setminus \{O_q'\}$ by exploiting the fact that, thanks to constraints (2), all variables w_q^{kl} are equal to 0 when $y_q=0$.

This leads to the following formulation denoted CCFRLM2:

$$\begin{cases} Max \sum\limits_{q\in\mathcal{Q}} f_q y_q & (19) \\ s.t. \quad \sum\limits_{k\in\mathcal{N}_{lq}^b} g_{kl}^q w_{kl}^q \leq \alpha \qquad \forall q\in\mathcal{Q},\ l\in\mathcal{N}_q\setminus\{O_q'\} & (20) \\ \quad (2)-(3), (5)-(8) \end{cases}$$

Constraints (20) state that, in case a trip q is covered, each node $l \in \mathcal{N}_q \setminus \{O_q'\}$ should be assigned to a node $k \in \mathcal{N}_{lq}^b$ such that the probability of running out of fuel when traveling from node k to node l is less than the risk parameter α. Constraints (20) are inactive in case q is not covered as all the assignment variables w_q^{kl} corresponding to trip q will be equal to 0.

5 Tabu search heuristic approach for the stochastic problem

Solving the stochastic flow refueling location models described in Section 4 using a commercial MILP solver is possible for small to medium size instances. However, when the size of the network increases, the computation time required to get an optimal solution might become prohibitively long. This is why we develop a tabu search heuristic approach to obtain good quality solutions in shorter computation times. Namely, tabu search is known to provide good results for generic facility location problems (see e.g. [1]) and to be easy to develop and implement. Moreover, such a heuristic method was successfully used in [24] for locating EV battery swapping stations.

In the following, we first describe the main principles used for building an initial solution for the heuristic algorithm and for evaluating each solution in terms of (expected) flow coverage. We then present the detailed tabu search procedure for solving the expected and chance constrained flow refueling location problems.

1. Building an initial solution

   The procedure of building an initial solution seeks to obtain a set of p stations that provide a good coverage of the network. To this end, the trips are considered in the decreasing order of their flow value f_q. The stations to be opened are selected in a way to cover first the trips with the largest flows. The procedure is stopped when the required number p of stations to be opened is reached.

   As explained in Section 4.2, in the chance constrained stochastic model, a trip is said to be covered if the probability of coverage (i.e. that the distance between any two consecutive stations built on q does not exceed the vehicle range) is higher than $1-\alpha$. To cover a trip q, we start by building a station at the origin O_q, then we greedily add stations on the shortest path between O_q and D_q in such a way that the probability of traveling between any two consecutive stations without running out of fuel is higher than $1-\alpha$.

   However, in the expected flow refueling location model, there is no minimum probability of coverage, since the aim is to maximize the expected flow covered. Therefore, we set this minimum probability as an input parameter when we build the initial solution and the procedure of building an initial set of p stations becomes similar to the one used for the chance constrained model. At the end of the procedure, we evaluate the objective function for the obtained solution, by checking the coverage (or probability of coverage) for all trips.

2. Evaluating the objective function for a given solution

   In order to evaluate the objective function for a given solution, we iterate over the trips to check which of them are covered and calculate the total (or expected) flow covered. To know if a trip q is covered, we proceed as follows: for each cycle segment $[k,l]$ of q, we first calculate the probability of coverage $(1-g^q_{kl})$ defined in Section 4.2. For the CCFRLM, the trip is covered if this probability is higher than $1-\alpha$ for all cycle segments of the trip. In this case, we add the flow of this trip to the objective value, otherwise we check the next trip. For the EFRLM, we calculate z_q the probability of coverage of the trip q as the minimum of the probabilities of coverage of all its cycle segments, then we update the objective value by adding $f_q z_q$.

The tabu search procedure used for solving the expected and chance-constrained problems is described in Algorithm 1. The algorithm starts by building an initial solution and by setting the best feasible solution Sol_{best} to the initial solution and the best objective value Obj_{best} to the initial objective value. Then, the procedure seeks to improve this solution iteratively by exploring its neighborhood. Each iteration of the algorithm comprises two steps. Step1 consists in selecting a station s_1 to be opened among the $|\mathcal{N}|-p$ closed stations. s_1 should not be tabu (not recently closed) and should lead to the highest (expected) coverage among all possible openings. This step leads to improving the (expected) coverage but the solution obtained is not feasible since it involves $p+1$ stations. For this reason, at the end of this step, we update the tabu list but we do not update the best solution Sol_{best}. Step2 consists in selecting a station s_2 to be closed among the $p+1$ opened stations. s_2 should not be tabu (not recently opened) and should lead

to the highest (expected) coverage among all possible closings. Here, we consider an aspiration criterion that consists in accepting to close a station that is tabu but leads to improving the best objective value. At the end of step2, we update the tabu list and we update the best solution Sol_{best} if it is improved after the last closing, otherwise, we increment the number of iterations without improvements. The tabu search procedure stops when the number of iterations without improvement of Obj_{best} reaches a maximum limit. Notice that in this procedure, we need two tabu lists, one for the recently opened stations and one for the recently closed stations.

Algorithm 1: Tabu search solution procedure for the stochastic flow refueling location problems

Data:
Sol_{init}: Initial solution (set of p stations);
Obj_{init}: Value of the objective function for the initial solution;
Sol_{best}: Current best feasible solution (set of p stations);
Obj_{best}: Value of the objective function for the current best feasible solution;
$Ntabu$: Size of tabu lists;
$tabuClosedList$: List of the stations closed in the last Ntabu iterations ;
$tabuOpenedList$: List of the stations opened in the last Ntabu iterations ;
$nbIterWithoutImpr$: Number of iterations of the tabu search without improvement of Obj_{best};
$maxWithoutImpr$: Maximum allowed value for $nbIterWithoutImpr$ (stopping criterion);

Algorithm:
Build an initial solution Sol_{init};
Initialize the current best solution: $Sol_{best} = Sol_{init}$ and $Obj_{best} = Obj_{init}$;
Initialize the number of iterations without improvement: $nbIterWithoutImpr = 0$;
while $nbIterWithoutImpr < maxWithoutImpr$ **do**
  Step1: Select a station s_1 to be opened among the $|\mathcal{N}| - p$ closed stations, such as $s_1 \notin tabuClosedList$ and opening s_1 leads to the highest (expected) coverage among all possible openings;
  if *no stations can be opened (all openings are tabu)* **then**
    Break, END TABU SEARCH
  else
    Add s_1 to $tabuOpenedList$;
  end
  Step2: Select a station s_2 to be closed among the $p+1$ opened stations such that closing s_2 leads to the highest (expected) coverage among all possible closings (we consider that closing a station s is possible if $s \notin tabuOpenedList$ or if we have the following aspiration criterion: $s \in tabuOpenedList$ but leads to improving the current best objective value Obj_{best});
  if *no stations can be closed* **then**
    Break, END TABU SEARCH
  else
    Add s_2 to $tabuClosedList$;
    if *closing s_2 leads to a better (expected) coverage than the current Obj_{best} value* **then**
      Update Obj_{best} and Sol_{best};
      $nbIterWithoutImpr = 0$
    else
      $nbIterWithoutImpr++$
    end
  end
end

6 Computational results

We now discuss the results of the numerical experiments carried out in order to assess the performance of the exact and heuristic solution approaches presented in the previous section. Subsection 6.1 describes the real-life instances used in the tests as well as the procedure employed to generate random instances. In Subsection 6.2, we compare formulation FRLM1 proposed in [3] and formulation FRLM2 developed in the present paper for the deterministic flow refueling location problem. Similarly, in Subsection 6.3, we compare formulation EFRLM1 (respectively CCFRLM1) proposed in [3] and formulation EFRLM2 (respectively CCFRLM2) developed in the present paper for the expected (respectively chance constrained) flow refueling location problem. Finally, Subsection 6.4 analyzes the computation times and optimality gaps for the tabu search heuristic.

All solution algorithms were coded in C++ language. The MILP models were implemented using the Concert Technology and solved using ILOG CPLEX version 12.6.2. In order to analyze the performance of the solution methods with other commercial solvers, a sample of random instances is also solved using XPRESS BCL version 4.8.14 provided in FICO XPRESS 8.5.6. The time limit was set to 10 hours for both solvers. All tests were carried out on a PC with Intel i5-3210M Core 2 Duo (2.50 GHz) with 8GB of RAM, running under Windows 10.

6.1 Problem instances

6.1.1 Random instances

In our numerical experiments, we use road networks which are randomly generated according to a procedure similar to the one proposed in [2]. We first generate $|\mathcal{N}|$ nodes whose coordinates are randomly chosen within $[1, 1000]^2$ according to a uniform distribution. The traveling distance between each pair of nodes is computed as the Euclidean distance. We then apply Kruskal algorithm to determine the minimum spanning tree of size $|\mathcal{N}| - 1$: all arcs belonging to this spanning tree are added to the arc set $\mathcal{A}$. We also select $|\mathcal{N}|$ additional arcs to be added to $\mathcal{A}$: these arcs are the shortest potential arcs not yet added to $\mathcal{A}$ and such that the degree of each node stays below four. We then randomly select M origin-destination nodes among the nodes belonging to $\mathcal{N}$. This gives the set of $|\mathcal{Q}| = \frac{M(M-1)}{2}$ trips to be covered. The shortest path $(\mathcal{N}_q, \mathcal{A}_q)$ corresponding to trip q is determined using Dijkstra algorithm. The population at each origin/destination node is randomly generated within $[1, 10000000]$ according to a uniform distribution. The flow f_q of EVs on each trip q is determined using the gravity model [4]: $f_q = \frac{P_q^O P_q^D}{d(O_q, D_q)^2}$ where P_q^O and P_q^D are the populations at the origin and destination nodes, respectively, and $d(O_q, D_q)$ is the length of the path between the origin O_q and the destination D_q.

The deterministic range R is set to 250km. The random range $R(\omega)$ is assumed to follow a Gamma distribution, with a shape parameter κ of 50 and a scale parameter θ of 5. For the chance constrained stochastic problem, the risk parameter α is set equal to 5 %. These assumptions are in line with the settings used in the paper of De Vries and Duijzer [3].

We consider two problem dimensions: $(|\mathcal{N}| = 100, M = 50)$ and $(|\mathcal{N}| = 200, M = 100)$, which correspond to 1225 (resp. 4950) trips. We randomly generate 5 instances of the network and set of trips to be covered. For each of these 10 network instances, we consider 9 possible values for the number of stations p: $p \in \{1, 2, 3, 4, 5, 10, 15, 20, 25\}$, resulting in a set of 90 instances. In the representation of each test instance, we include first $|\mathcal{N}|$, then M, then p, for example, the instance $N100M50p10$ deals with a network of 100 nodes, 50 origin/destination nodes and 10 stations to be opened.

6.1.2 Real-life instances

We also study two real-life road networks: the California network used in [25] and the Florida network used in [16]. The California network consists of 339 nodes and 1167 paths whereas the Florida network consists of 302 nodes and 2701 paths. In both cases, the volume of the flow on each path is computed using the gravity model and the number of stations to be opened is varied in $\{5, 10, 15, 20, 25\}$. For the California network, we consider two values for the deterministic range R: 100km and 150km. Accordingly, we set the parameters for the Gamma distribution $R(\omega)$ to $\kappa = 50$, $\theta = 2$ and $\kappa = 50$, $\theta = 3$. For the California network, we consider two values for the deterministic range R: 50km and 100km. Accordingly, we set the parameters for the Gamma distribution $R(\omega)$: $\kappa = 50$, $\theta = 1$ and $\kappa = 50$, $\theta = 2$. There is one instance for each network, each expected range and each number of stations, which results in a set of 20 instances. Instances are refereed to as $Net.R.p$ where $Net \in \{CA, FL\}$ relates to the network, R to the value of the expected range and p to the number of stations to be opened.

6.2 Results for the deterministic problem

In this section, we analyze the numerical performance of formulations FRLM1 proposed in [3] and FRLM2 described in Section 3 for the deterministic FRLP. Table 2 (resp. 3) show the results obtained using CPLEX on randomly generated networks (resp. on real-life networks) for both models. They display the integrality gap computed as the relative difference between the optimal integer feasible value and the linear relaxation value, the CPU time in seconds (we set the computation time limit to 10 hours), the number of Branch & Bound nodes explored by the algorithm within the time limit and the residual gap defined as the relative difference between the best integer feasible solution value and the best upper bound found within the time limit.

First, we observe that the average CPU time increases with the network size. This might be explained by the fact that the formulation size increases when $|\mathcal{N}|$ and M increase. For instance, for the randomly generated instances, the number of variables for both formulations is on average 158113 for the $N100M50pXX$ instances and 1091989 for the $N200M100pXX$ instances. As for the constraints, formulation FRLM1 (resp. FRLM2) involves on average 50291 (resp. 173285) constraints for the $N100M50pXX$ instances and 273407 (resp. 1177975) constraints for the $N200M100pXX$ instances. We also observe that the average CPU time tends to increase with the value of p, which might be explained by the fact that the number of possible combinations for stations deployment increases as long as $p \leq |\mathcal{N}|/2$. Note that there are exceptions for small values of p using FRLM2 (see Table 2) and for $p = 20$ with California instance using FRLM1 (see Table. 3).

Second, the results show that the average computation time is significantly reduced on random instances (from 11643s to 422s) and on real life instances (from 24879s to 790s) when using formulation FRLM2. Moreover, with formulation FRLM2, all 110 considered instances could be solved to optimality within the time limit while only 90 could be solved to optimality with formulation FRLM1. As seen from the residual gap reported in column 5 of Table 2, for the largest instances, the solutions obtained when using FRLM1 after 10h of computation time are of rather poor quality.

The better numerical performance observed with formulation FRLM2 could be explained by the fact that contrary to FRLM1, the proposed formulation does not include any big-M constraints. This leads to a tighter linear relaxation: on random (resp. real-life) instances, the average integrality gap is namely 20.9% (resp. 10.1%) with FRLM2 while it is 73.3% (resp. 22.7%) with FRLM1. As a consequence, the average number of Branch & Bound nodes explored by the algorithm before

finding an optimal solution is significantly reduced from 1554 (resp. 6726) with FRLM1 to 10 (resp. 38) with FRLM2. Hence, despite the fact that formulation FRLM2 has an order of magnitude more constraints than formulation FRLM1, its better tightness allows the mathematical solver CPLEX to more efficiently solve the deterministic variant of the problem.

	FRLM1				FRLM2			
Test instance	Integrality Gap (%)	CPU (s)	B&B nodes	Residual Gap (%)	Integrality Gap (%)	CPU (s)	B&B nodes	Residual Gap (%)
$N100M50p1$	323.8	25	0	0.0	62.3	24	0	0.0
$N100M50p2$	174.7	96	54	0.0	54.9	15	0	0.0
$N100M50p3$	107.6	156	159	0.0	43.3	14	0	0.0
$N100M50p4$	73.4	308	190	0.0	34.1	12	0	0.0
$N100M50p5$	53.5	457	155	0.0	27.9	14	0	0.0
$N100M50p10$	20.5	3693	2518	0.0	15.9	21	4	0.0
$N100M50p15$	9.2	7568	4033	0.0	8.3	31	12	0.0
$N100M50p20$	3.8	12008	5862	0.1	3.7	40	26	0.0
$N100M50p25$	1.9	17674	11521	0.0	1.9	42	20	0.0
$N200M100p1$	196.2	502	0	0.0	12.1	1947	0	0.0
$N200M100p2$	111.8	1814	246	0.0	19.9	265	0	0.0
$N200M100p3$	76.0	4000	90	0.0	17.3	380	0	0.0
$N200M100p4$	58.5	6290	422	0.0	16.6	409	0	0.0
$N200M100p5$	47.4	11829	972	0.0	15.8	431	0	0.0
$N200M100p10$	26.4	35157	1438	8.9	14.4	648	1	0.0
$N200M100p15$	17.4	36000	313	12.4	12.5	782	4	0.0
$N200M100p20$	11.1	36000	0	#	9.4	1170	30	0.0
$N200M100p25$	6.4	36000	0	#	5.9	1352	86	0.0
Average	**73.3**	**11643**	**1554**	#	**20.9**	**422**	**10**	**0.0**

Table 2: FRLM1 and FRLM2 comparison on random instances (average of 5 replications, # denotes the case where the residual gap could not be computed due to the fact that the best integer solution found has a value (nearly) equal to 0)

6.3 Results for the stochastic problem

This section compares the formulations discussed in Section 4 for the stochastic flow refueling location problem. Tables 4 and 5 display the results obtained using CPLEX on randomly generated instances for the expected flow refueling location problem (EFRLP) and the chance constrained flow refueling location problem (CCFRLP). The results obtained for both problems on the real-life instances are summarized in Table 6 .

Regarding the EFRLP, results from Table 4 and Table 6 show that on average, the proposed formulation EFRLM2 performs better than the previously published formulation EFRM1. Namely, the average computation time on random instances is reduced from 9522s when using EFRLM1 to 5753s when using EFRLM2 while the reduction is more significant on real-life instances (from 12716s to 2646s). Moreover, 85 random instances and 20 real-life instances could be solved to optimality within the time limit with formulation EFRLM2 while only 73 random instances and 18 real-life instances could be solved to optimality with formulation EFRLM1. However, we note that formulation EFRLM2 seems to be more performant than formulation EFRLM1 for large values of p only (typically more than 5 stations). This might be explained by the fact that both formulations involve similar knapsack constraints (see constraints (17) linking the trip coverage variables z_q with the assignment variables w_q^{kl} in EFRLM2). Hence, the only formulation

	FRLM1				FRLM2			
Test instance	Integrality Gap (%)	CPU (s)	B&B nodes	Residual Gap (%)	Integrality Gap (%)	CPU (s)	B&B nodes	Residual Gap (%)
CA.100.5	35	2491	1557	0.0	19	193	0	0.0
CA.100.10	9	36000	20888	1.2	6	197	0	0.0
CA.100.15	3.3	36000	22856	1.2	2.8	547	97	0.0
CA.100.20	1.2	36000	27749	1.6	1.2	727	75	0.0
CA.100.25	0.1	36000	3197	0.4	0.1	1066	232	0.0
CA.150.5	19	1240	1103	0.0	8	87	0	0.0
CA.150.10	5	36000	22320	0.0	3	574	36	0.0
CA.150.15	1.3	36000	35445	0.3	1,1	577	19	0.0
CA.150.20	0.1	29253	9667	0.0	0.1	1085	124	0.0
CA.150.25	0	36000	762	0.0	0	9251	58	0.0
FL.50.5	112	2521	135	0.0	39	14	0	0.0
FL.50.10	56	4526	252	0.0	25	15	0	0.0
FL.50.15	35	19798	2860	0.0	18	15	0	0.0
FL.50.20	27	36000	5979	0.8	16	25	0	0.0
FL.50.25	21	36000	4200	6.5	15	42	0	0.0
FL.100.5	59	1051	43	0.0	11	93	0	0.0
FL.100.10	30	4704	1280	0.0	11	107	0	0.0
FL.100.15	19	36000	8267	0.0	11	218	0	0.0
FL.100.20	13	36000	4556	2.9	9	385	50	0.0
FL.100.25	8	36000	6705	3.1	6	575	71	0.0
Average	**22.7**	**24879**	**6726**	**0.9**	**10.1**	**790**	**38**	**0.0**

Table 3: FRLM1 and FRLM2 comparison on real-life instances

strengthening obtained when using formulation EFRM2 comes from the integration of a second set of constraints (constraints (16)) to link variables z_q and w_q^{kl}.

As for the CCFRLP, we observe a significant reduction of the average computation time (from 8086s to 201s for random instances and from 10577s to 226s for real-life instances) when using formulation CCFRLM2 rather than formulation CCFRLM1. Moreover, all random and real-life instances could be solved to optimality with formulation CCFRLM2 while 16 random instances and 2 real-life instances could not be solved to optimality with formulation CCFRLM1. This strong improvement in numerical efficiency might be explained by the fact that big-M constraints in formulation CCFRLM1 are replaced by knapsack constraints (20) in formulation CCFRLM2.

Finally, in order to check whether the relative performance of the MILP models might depend on the mathematical programming solver, we conduct some experiments using the XPRESS-MP solver. We solve formulations EFRLM1, EFRLM2, CCFRLM1 and CCFRLM2 for a sample of 15 random instances used in Tables 4 and 5 (namely 5 replications of each of $N100M50p15$, $N100M50p20$ and $N100M50p25$). The average computation time for each problem, each formulation and each set of 5 instances is reported in Table 7. By comparing the results in Table 7 with those in Tables 4 and 5, we first note that XPRESS-MP performs better than CPLEX when using formulations EFRLM1 and CCFRLM1 while CPLEX performs better with the new formulations EFRLM2 and CCFRLM2. Moreover, results from Table 7 show that, similarly to what is observed with CPLEX solver, the proposed formulations perform better than the existing ones with XPRESS-MP. Namely, the computation times obtained with formulations EFRLM2 and CCFRLM2 are shorter than the ones obtained with formulations EFRLM1 and CCFRLM1. Furthermore, the performance improvement observed for the chance constrained problem is more significant than the one observed for the expected flow refueling location problem. This confirms that the proposed formulations are likely to outperform the existing ones in any generic MILP

solver.

Test instance	CPU (s)	EFRLM1 B&B nodes	Residual Gap (%)	CPU (s)	EFRLM2 B&B nodes	Residual Gap (%)
$N100M50p1$	18	0	0.0	111	0	0.0
$N100M50p2$	36	21	0.0	55	0	0.0
$N100M50p3$	65	43	0.0	72	3	0.0
$N100M50p4$	100	198	0.0	83	20	0.0
$N100M50p5$	181	210	0.0	83	18	0.0
$N100M50p10$	583	484	0.0	165	0	0.0
$N100M50p15$	1595	2594	0.0	487	163	0.0
$N100M50p20$	4815	2491477	0.0	597	66	0.0
$N100M50p25$	11008	8835	0.2	609	263	0.0
$N200M100p1$	371	0	0.0	5033	0	0.0
$N200M100p2$	1465	0	0.0	1762	0	0.0
$N200M100p3$	2428	0	0.0	1831	0	0.0
$N200M100p4$	4218	0	0.0	16430	0	34.0
$N200M100p5$	3730	0	0.0	3159	0	0.0
$N200M100p10$	32776	645	#	8212	88	0.0
$N200M100p15$	36000	629	6.4	17243	126	0.0
$N200M100p20$	36000	7	6.6	21237	150	0.2
$N200M100p25$	36000	93	5.9	26385	174	0.6
Average	**9522**	**139180**	#	**5753**	**59**	**2.0**

Table 4: EFRLM1 and EFRLM2 comparison on random instances (average of 5 replications, # denotes the case where the residual gap could not be computed due to the fact that the best integer solution found has a value (nearly) equal to 0)

Test instance	CPU (s)	CCFRLM1 B&B nodes	Residual Gap (%)	CPU (s)	CCFRLM2 B&B nodes	Residual Gap (%)
$N100M50p1$	20	0	0.0	14	0	0.0
$N100M50p2$	60	12	0.0	7	0	0.0
$N100M50p3$	71	12	0.0	9	0	0.0
$N100M50p4$	123	17	0.0	9	0	0.0
$N100M50p5$	141	14	0.0	9	0	0.0
$N100M50p10$	426	49	0.0	11	0	0.0
$N100M50p15$	775	346	0.0	13	0	0.0
$N100M50p20$	1152	412	0.0	16	0	0.0
$N100M50p25$	1520	317	0.0	17	0	0.0
$N200M100p1$	442	0	0.0	1120	0	0.0
$N200M100p2$	2144	5	0.0	216	0	0.0
$N200M100p3$	2524	39	0.0	158	0	0.0
$N200M100p4$	2663	45	0.0	166	0	0.0
$N200M100p5$	4210	87	0.0	254	2	0.0
$N200M100p10$	22008	2093	1.9	268	0	0.0
$N200M100p15$	35268	1040	6.3	346	2	0.0
$N200M100p20$	36000	168	9.0	454	20	0.0
$N200M100p25$	36000	0	33.7	534	7	0.0
Average	**8086**	**259**	**2.8**	**201**	**2**	**0.0**

Table 5: CCFRLM1 and CCFRLM2 comparison on random instances (average of 5 replications)

	EFRLM1		EFRLM2		CCFRLM1		CCFRLM2	
Test instance	CPU (s)	Residual Gap (%)	CPU (s)	Residual Gap (%)	CPU (s)	Residual Gap (%)	CPU (s)	Residual Gap (%)
CA.100.5	934	0.0	848	0.0	1440	0.0	66	0.0
CA.100.10	1530	0.0	1049	0.0	2680	0.0	60	0.0
CA.100.15	4442	0.0	2381	0.0	4896	0.0	237	0.0
CA.100.20	36000	0.0	4877	0.0	11363	0.0	413	0.0
CA.100.25	36000	0.0	4877	0.0	36000	0.0	416	0.0
CA.150.5	341	0.0	1097	0.0	771	0.0	83	0.0
CA.150.10	1272	0.0	1231	0.0	3313	0.0	181	0.0
CA.150.15	2162	0.0	1306	0.0	5158	0.0	429	0.0
CA.150.20	7667	0.0	3460	0.0	25885	0.0	847	0.0
CA.150.25	18772	0.0	10157	0.0	5983	0.0	401	0.0
FL.50.5	822	0.0	106	0.0	1204	0.0	7	0.0
FL.50.10	3017	0.0	140	0.0	1979	0.0	7	0.0
FL.50.15	5867	0.0	157	0.0	2879	0.0	10	0.0
FL.50.20	9763	0.0	215	0.0	9993	0.0	9	0.0
FL.50.25	36000	0.0	671	0.0	8317	0.0	7	0.0
FL.100.5	649	0.0	299	0.0	1095	0.0	295	0.0
FL.100.10	2548	0.0	1420	0.0	2745	0.0	113	0.0
FL.100.15	14530	0.0	3156	0.0	13830	0.0	117	0.0
FL.100.20	36000	2.0	6867	0.0	36000	1.0	354	0.0
FL.100.25	36000	2.4	8615	0.0	36000	2.0	459	0.0
Average	**12716**	**0.2**	**2646**	**0.0**	**10577**	**0.1**	**226**	**0.0**

Table 6: Comparison of stochastic problem formulations on real-life instances

Test instance	EFRLM1(s)	EFRLM2 (s)	CCFRLM1 (s)	CCFRLM2 (s)
$N100M50p15$	1210	806	344	24
$N100M50p20$	1474	608	311	29
$N100M50p25$	2890	783	484	15
Average	**1858**	**732**	**380**	**23**

Table 7: Comparison of XPRESS CPU (s) for the stochastic problems on random instances (average of 5 replications)

6.4 Performance of the tabu search procedure

As shown by the computational results presented in SubSection 6.3, the CPU times needed to solve to optimality the MILP formulations corresponding to the stochastic problem remain long for large-size instances. For instance, Table 4 shows that even when using the new formulation EFRLM2, the average computation time needed to solve the problem on random instances with $|\mathcal{N}| = 200$ nodes, $M = 100$ origin-destination nodes and $p \geq 15$ is around 5 to 7 hours. A heuristic solution approach could thus be useful to obtain good quality solutions in shorter running times. In this subsection, we study the performance of the tabu search heuristic algorithm described in Section 5. We use the tabu search procedure to solve the stochastic problems with the following settings: tabu list size $Ntabu = 5$, maximum allowed number of iterations without improvement of the objective value $maxWithoutImpr = 10$ and minimum probability of coverage set equal to 60% when building the initial solution for the expected flow refueling location problem (for the chance constrained problem, the minimum probability of coverage is set by default to 95% since α is equal to 5%).

Tables 8 and 9 summarize the results obtained with the heuristic algorithm on the random

and real-life instances of the stochastic problems and compare them with the ones obtained while solving the problem to optimality using CPLEX with the MILP formulation displaying the best performance on the corresponding instance set. In the tables, we report the average tabu gap, which is defined as the relative difference between the best solution value found by the heuristic algorithm and the optimal solution value (or the best feasible solution if the optimal solution was not found by the solver within the time limit), together with the CPU times for the tabu search heuristic and the mathematical solver.

Results from Tables 8 and 9 first show that the tabu heuristic is capable of providing near-optimal solutions within short computation times. Namely the average gap of the tabu heuristic on random instances is 0.9% for the EFRLP and 0.8% for the CCFRLP while the average computation time is 35s for the EFRLP and 23s for the CCFRLP. Similarly, on real-life networks, the heuristic provides good quality solutions: the average tabu gap is 1.3% for the EFRLP and 0.7% for the CCFRLP while the average computation time does not exceed 30s for both problems.

	EFRLP			CCFRLP		
Test instance	Tabu gap (%)	Tabu CPU (s)	Best CPLEX CPU (s)	Tabu gap (%)	Tabu CPU (s)	Best CPLEX CPU (s)
$N100M50p1$	0.0	1	18	0	1	14
$N100M50p2$	2.7	2	36	0	1	7
$N100M50p3$	4.5	2	65	0.1	2	9
$N100M50p4$	1.3	3	83	0	5	9
$N100M50p5$	2.0	7	83	0	12	9
$N100M50p10$	0.2	9	165	4.2	9	11
$N100M50p15$	0.2	14	487	0.3	11	13
$N100M50p20$	0.3	20	597	0.4	11	16
$N100M50p25$	0.1	24	609	0.6	13	17
$N200M100p1$	2.2	7	371	0	8	1120
$N200M100p2$	0	13	1465	0	12	216
$N200M100p3$	0.3	16	1831	0.2	15	158
$N200M100p4$	0.3	23	4219	1.7	20	166
$N200M100p5$	0.3	54	3159	0.2	44	254
$N200M100p10$	0.8	58	8212	0.8	64	268
$N200M100p15$	0.7	90	17243	1.4	73	346
$N200M100p20$	0.8	102	21237	1.2	55	454
$N200M100p25$	0.6	184	26385	2.5	64	534
Average	**0.9**	**35**	**4792**	**0.8**	**23**	**201**

Table 8: Performance of the tabu heuristic for the stochastic problems on random instances (average of 5 replications)

We also note that, for the EFRLP, the tabu search heuristic approach compares well with the mathematical solver as it leads to near-optimal solutions within a significantly reduced computation time. Namely, over all instances, the average computation time is decreased from 4792s with the solver to 35s with the heuristic. This reduction is even more substantial for the largest instances (see the random instance sets $N200M100p15$, $N200M100p20$ and $N200M100p25$) for which the average computation time is decreased from 6h with the solver to 2min with the heuristic.

However, for the CCFRLP, the tabu search heuristic approach compares less favorably with the mathematical solver. Thus, over the 110 considered instances, the average computation time is decreased from 205s with the solver to 24s with the heuristic. Note that this less favorable comparison is not due to the fact that the performance of the heuristic is poorer on the CCFRLP

Test instance	EFRLP Tabu gap (%)	EFRLP Tabu CPU (s)	EFRLP Best CPLEX CPU (s)	CCFRLP Tabu gap (%)	CCFRLP Tabu CPU (s)	CCFRLP Best CPLEX CPU (s)
CA.100.5	1.7	8	848	0.8	11	66
CA.100.10	2.9	10	1049	0.8	18	60
CA.100.15	1.2	24	2381	0.7	15	237
CA.100.20	0.5	34	4877	0.8	25	413
CA.100.25	0.9	17	4877	0.6	18	416
CA.150.5	0.0	8	341	0.0	25	83
CA.150.10	0.0	15	1231	0.4	13	181
CA.150.15	0.6	13	1306	0.2	26	429
CA.150.20	0.3	13	3460	0.8	12	847
CA.150.25	0.0	32	10157	0.1	21	401
FL.50.5	1.9	13	106	0.0	25	7
FL.50.10	4.2	16	140	0.0	28	7
FL.50.15	0.0	26	157	2.6	52	10
FL.50.20	2.0	28	215	2.0	37	9
FL.50.25	2.3	26	671	1.7	53	7
FL.100.5	2.8	15	299	0.0	23	295
FL.100.10	0.9	22	1420	0.0	30	113
FL.100.15	1.3	40	3156	0.2	46	117
FL.100.20	1.0	31	6867	0.8	67	354
FL.100.25	2.4	52	8615	2.0	58	459
Average	**1.3**	**22**	**2609**	**0.7**	**30**	**226**

Table 9: Performance of the tabu heuristic for the stochastic problems on real-life instances

than on the EFRLP. It is rather explained by the drastic decrease in the computation times needed by CPLEX solver to solve the new formulation CCFRLM2. Accordingly, for the CCFRLP, using CPLEX solver with formulation CCFRLM2 seems to be the best option.

7 Conclusion and future research

In this paper, we considered the flow refueling location problem under stochastic driving range and studied two stochastic programming based formulations for this problem: a first one maximizing the expected EV flow coverage and a second one based on joint chance constraints. We focused on decreasing the computation times needed to solve the resulting large-size combinatorial optimization problems and proposed two new efficient solution approaches. The first one uses a new location-allocation type model for this problem and leads to a strong MILP formulation. Our computational experiments carried out on large-size random instances as well as on real-life instances show that a significant improvement in numerical performance can be obtained when using the new model. Namely, the average CPU time needed to solve the chance constrained model is reduced by 95%. In the case of expected flow coverage maximization, the improvement in performance was obvious for large-size networks but not for small-size ones. Moreover, the CPU time for networks with a large number of stations to be opened remained rather high (around 7-9 hours for random instances). Therefore, we proposed a tabu search-based heuristic in order to obtain good quality solutions within shorter computation times. The heuristic showed excellent numerical performance for the expected flow coverage maximization problem, with low optimality gaps and short running times.

In our models, we assumed that for any realization of the random conditions in the road network, the value of the EV range is the same for the whole network. Therefore, in terms of future research, it would be interesting to relax this assumption and to study a more realistic case where the driving range is different on each cycle segment of a path. This relaxation will raise the problem of computing the joint probability when evaluating the coverage on each path of the network. Other future research directions include the introduction of a limited capacity for the charging stations as well as the dynamic aspect of the problem.

References

[1] M. A. Arostegui, S. N. Kadipasaoglu, and B. M. Khumawala. An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics*, 103:742–754, 2016.

[2] I. Capar, M. Kuby, V. J. Leon, and Y.-J. Tsai. An arc cover-path-cover formulation and strategic analysis of alternative-fuel station locations. *European Journal of Operational Research*, 227:142—151, 2013.

[3] H. de Vries and E. Duijzer. Incorporating driving range variability in network design for refueling facilities. *Omega*, 69:102–114, 2017.

[4] A. Fotheringham and M. O'Kelly. Spatial interaction models: Formulations and applications. *Kluwer Academic Publishers*, 1989.

[5] S. Funke, A. Nusser, and S. Storandt. Placement of loading stations for electric vehicles: no detours necessary! *Journal of Artificial Intelligence Research*, 53:633–658, 2015.

[6] A. Gagarin and P. Corcoran. Multiple domination models for placement of electric vehicle charging stations in road networks. *Computers and Operations Research*, 96:69–79, 2018.

[7] M. J. Hodgson. A flow capturing location-allocation model. *Geographical analysis*, 22:270–279, 1990.

[8] M. Hosseini and S. A. MirHassani. A heuristic algorithm for optimal location of flow-refueling capacitated stations. *International Transactions on Operational Research*, 24:1377–1403, 2012.

[9] M. Hosseini and S. A. MirHassani. Refueling station location problem under uncertainty. *Transportation Research Part E*, 84:101–116, 2015.

[10] Y. Huang, S. Li, and Z. S. Qian. Optimal deployment of alternative fueling stations on transportation networks considering deviation paths. *Networks and Spatial Economics*, 15: 183–204, 2015.

[11] International Energy Agency. Transport energy and CO2: moving toward sustainability. Technical report, 2009.

[12] International Energy Agency. Global EV outlook 2017. Technical report, 2017.

[13] J. Jung, J. Y. J. Chow, R. Jayakrishnan, and J. Y. Park. Stochastic dynamic itinerary interception refueling location problem with queue delay for electric taxi charging stations. *Transportation Research Part C*, 40:123–142, 2014.

[14] J. G. Kim and M. Kuby. A network transformation heuristic approach for the deviation flow refueling location model. *Computers and Operations Research*, 40:1122–1131, 2013.

[15] M. Kuby and S. Lim. The flow-refueling location problem for alternative-fuel vehicles. *Socio-Economic Planning Sciences*, 39:125–145, 2005.

[16] M. Kuby, L. Lines, R. Schultz, Z. Xie, J. G. Kim, and S. Lim. Optimization of hydrogen stations in Florida using the flow-refueling location model. *International Journal Of Hydrogen Energy*, 34:6045–6064, 2009.

[17] C. Lee and J. Han. Benders-and-price approach for electric vehicle charging station location problem under probabilistic travel range. *Transportation Research Part B*, 106:130–152, 2017.

[18] S. Li and Y. Huang. Development of electric vehicle charging corridor for South Carolina. *International Journal of Transportation Science and Technology*, 4:395–411, 2015.

[19] I. Y. Mak, Y. Rong, and Z. J. M. Shen. Infrastructure planning for electric vehicles with battery swapping. *Management Science*, 59:1557–1575, 2013.

[20] S. A. MirHassani and R. Ebrazi. A flexible reformulation of the refueling station location problem. *Transportation Science*, 47:617–628, 2012.

[21] C. Upchurch and M. Kuby. A capacitated model for location of alternative-fuel stations. *Geographical analysis*, 41:89–106, 2009.

[22] US Department of Energy. National plug-in electric vehicle infrastructure analysis. Technical report, 2017.

[23] F. Wu and R. Sioshansi. A stochastic flow-capturing model to optimize the location of fast-charging stations with uncertain electric vehicle flows. *Transportation Research Part D*, 53: 354–376, 2017.

[24] J. Yang, F. Guo, and M. Zhang. Optimal planning of swapping/charging station network with customer satisfaction. *Transportation Research Part E*, 103:174–197, 2017.

[25] B. Yildiz, O. Arslan, and O. Ekin Karasan. A branch and price approach for routing and refueling station location model. *European Journal of Operational Research*, 248:815–826, 2016.