Les Cahiers du GERAD ISSN: 0711–2440

Electric vehicle fast-charging facility location with endogenous queuing in path selection

W. Wang, A. Dems, Y. Adulyasak, O. Arslan, J.-F. Cordeau

G–2026–11

March 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Electric vehicle fast-charging facility location with endogenous queuing in path selection

Weiquan Wang [a]

Amira Dems [b]

Yossiri Adulyasak [a]

Okan Arslan [a]

Jean-François Cordeau [a]

[a] *GERAD & HEC Montréal, Montréal (Qc), Canada, H3T 1J4*

[b] *Institut de recherche d'Hydro-Québec, Varennes (Qc), Canada, J3X 1S1*

weiquan.wang@hec.ca
dems.amira@hydroquebec.com
yossiri.adulyasak@hec.ca
okan.arslan@hec.ca
jean-francois.cordeau@hec.ca

March 2026
Les Cahiers du GERAD
G–2026–11

Abstract : As fast-charging demand grows, ensuring the operational resilience of electric vehicle (EV) infrastructure under congestion becomes a key challenge in large-scale transportation systems. We study a bilevel charging facility location problem in which EV drivers choose paths based on a disutility function incorporating travel time, charging stops, and queuing delays modeled via an $M/M/c$ system. We develop both arc-flow and path-based bilevel formulations and derive equivalent single-level mixed-integer linear programming (MILP) reformulations using strong duality and piecewise-linear approximations of queuing delays. To address the computational challenges arising from large-scale transportation networks, we propose an exact decomposition algorithm (DA) that iteratively solves a location master problem and an evaluation subproblem capturing path choice and congestion effects. Several acceleration strategies are introduced to improve convergence speed significantly. Computational results show that the proposed DA substantially outperforms the direct solution of the single-level MILP formulations using a commercial solver on small networks. Extensive experiments on large real-world networks from California and a case study in Québec demonstrate strong scalability, with most instances solved to optimality or near-optimality within reasonable time limits. Benchmark comparisons on a closely related problem from the literature further show that the proposed DA consistently outperforms the state-of-the-art algorithm in both solution quality and computational robustness.

Keywords : Facility location; electric vehicles; charging facilities; bilevel optimization

1 Introduction

The rapid growth of electric vehicle (EV) adoption is placing significant operational pressure on fast-charging infrastructure across North America. In California, zero-emission vehicles (ZEVs) accounted for 29.1% of all new vehicle sales between July and September 2025—the largest share on record—with more than 108,600 EVs sold in Q3 alone, representing a 30% increase over Q2.[1] As of September 2025, the state already had over 2.45 million ZEVs on its roads.[2] In Québec, over 404,000 electric and plug-in hybrid vehicles were in service as of June 2025, and this number is projected to reach two million by 2030. The province also operates more than 13,300 publicly accessible charging points, including over 2,200 fast-charging stations.[3] Fast-charging facilities—particularly Level-3 DC chargers capable of adding approximately 200–400 km of driving range within about 30 minutes—are crucial for long-distance EV travel.[4] However, when heavy EV flows concentrate at a limited number of fast-charging sites, especially along highway corridors, queuing delays are inevitable.

For EV drivers taking long-distance trips, multiple detour options may be available along the way. Their path choices depend not only on travel time and the number of required charging stops but also on the expected queuing times at charging facilities. Drivers tend to prefer routes with shorter travel times, fewer charging stops, and lower expected queuing delays (Xu and Meng 2020, Kınay, Gzara, and Alumur 2023). To model this behavior, we design a disutility function that jointly accounts for travel time, charging time, and queuing delays. To the best of our knowledge, this study is the first to explicitly incorporate queuing delays in EV path-selection disutility and to model the resulting feedback with charging facility location decisions. We model the expected queuing time at each charging node using an $M/M/c$ system, where EV arrivals follow a Poisson process, service times are exponential, and charging facilities operate under a single-line, first-come, first-served policy.

Importantly, EV drivers' path selections induce the charging demand at each charging node, which in turn affects the resulting queuing delays, and these delays influence the drivers' path selection. This creates an interplay between path selection decisions and congestion at charging nodes. Our objective is to determine an optimal deployment of charging facilities, under a given budget, that maximizes the amount of origin–destination (OD) flow served. To better capture real-world driving behavior and EV range anxiety, we consider multiple EV range settings and allow drivers to deviate from the shortest-travel-time path within a specified deviation threshold (Arslan et al. 2019, Park and Lee 2024).

The five main contributions of this work are as follows.

(i) We introduce a new bilevel charging facility location problem that explicitly models EV path selection under a disutility function incorporating travel time, charging time, and queuing delays modeled via an $M/M/c$ system.

(ii) We develop both arc-flow based and path-based bilevel formulations and derive mathematically equivalent single-level mixed-integer linear programming (MILP) reformulations using strong duality and piecewise-linear approximations of the $M/M/c$ queuing function.

(iii) We propose an exact decomposition algorithm (DA), which consists of a Refueling Station Location Problem with Routing (RSLP) as the master problem (MP) that repeatedly generates high-quality location vectors, and a subproblem (SP) that evaluates these vectors by accounting for path selection and queuing delays. We introduce new acceleration strategies that make the state-of-the-art RSLP solution method (Arslan et al. 2019), on average, ten times faster, allowing the MP to be solved within seconds.

(iv) Through extensive computational experiments, we demonstrate that the DA delivers order-of-magnitude speedups over MILP reformulations with Gurobi on small 25-node networks, and

[1] https://www.evinfrastructurenews.com/ev-networks/california-sets-ev-sales-record-in-q3-2025-ahead-of-tax-credit-expiry

[2] https://www.energy.ca.gov/news/2025-03/california-exceeds-178000-electric-vehicle-chargers

[3] https://www.quebec.ca/en/transports/electric-transportation/charging/charging-stations-anywhere

[4] https://www.tesla.com/support/charging

solves 91 of 120 large real-world California (339-node) and Québec (405-node) networks to optimality within two hours, while maintaining a low average optimality gap.

(v) We benchmark the proposed DA on the charging station location and sizing problem for electric vehicles under congestion (Kınay, Gzara, and Alumur 2023) and demonstrate that it consistently outperforms the state-of-the-art algorithm.

The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 describes the problem setting. Section 4 presents the two bilevel formulations. Section 5 derives the single-level MILP reformulations. Section 6 introduces the proposed decomposition algorithm. Section 7 reports the computational results and insights. Section 8 concludes the paper.

2 Literature review

The rapid adoption of electric vehicles (EVs) has created an urgent need for well-designed charging infrastructure to support large-scale electrification of transportation networks. The problem of charging facility location has therefore been extensively studied in the operations research and transportation science literature.

Early work on EV charging station planning was largely inspired by the Flow Refueling Location Model (FRLM), introduced by Kuby and Lim (2005). Capar et al. (2013) proposed arc-flow and path-flow logic, while MirHassani and Ebrazi (2013) proposed an expanded transformed network that drastically increases the solution efficiency of the FRLM. Subsequently, numerous studies have proposed extensions and solution methods for the FRLM (Chung and Kwon 2015, Zhang et al. 2016, Arslan and Karaşan 2016, Lee and Han 2017, Xu and Meng 2020, Park and Lee 2024).

An important extension of the FRLM is to allow vehicles to select paths that deviate from the shortest-distance route within a given tolerance threshold, capturing the fact that drivers may accept limited detours to access reliable refueling opportunities. This setting is commonly referred to as the RSLP with routing (RSLP-R), introduced by Yıldız, Arslan, and Karaşan (2016). To overcome the computational difficulties of solving large-scale instances, Arslan et al. (2019) and Göpfert and Bock (2019) independently proposed a branch-and-cut algorithm, in which the length-bounded cuts are used to improve computational efficiency. This length-bounded concept has been also applied in subsequent RSLP-R studies (Mahmutoğulları and Yaman 2023, 2024). Kınay, Gzara, and Alumur (2021) relaxed the deviation tolerance of drivers' path choices and considered a full-cover station location problem, aiming to minimize the station construction cost and the total en-route charging.

Note that the above studies do not consider station capacity and implicitly assume that a station can handle all charging demand passing through it. To the best of our knowledge, no existing study explicitly captures the feedback mechanism between path choices and congestion-induced queuing times at charging facilities. Zhang, Kang, and Kwon (2017) considered a capacitated fast-charging location problem within a multi-period framework, where the flow dynamically increases as the charging capacity along the paths expands. Xie et al. (2018) modeled station capacity by considering an $M/M/c$ queuing system and proposed requirements for the service level of the stations, but they did not consider queuing time in path selection. Anjos, Gendron, and Joyce-Moniz (2020) modeled the charging demand at stations within a multi-period framework by combining node-based and flow-based approaches, and considered that EV adoption varies with the availability of charging opportunities.

To capture realistic driver behavior, Lamontagne et al. (2023) adopted the simulation-based method of Paneque et al. (2021), developed a discrete choice stochastic model at two levels for the placement of charging stations, and reformulated into a maximum coverage problem to improve computational tractability. Legault and Frejinger (2025) proposed a simulation-based and submodularity-driven approach to competitive facility location under multinomial and mixed logit models, with an emphasis on competition for market share.

To capture drivers' travel behavior responses to location decisions, many studies consider bilevel location optimization problems. In He et al. (2018)'s research, the upper level determines station locations, and the lower level formulates a user equilibrium traffic assignment, where drivers select routes based on travel times and charging feasibility. Arslan, Jabali, and Laporte (2018) introduced the Evasive Flow Capturing Problem (EFCP), a non-cooperative bilevel setting relevant to enforcement and security applications. The upper level places enforcement facilities to minimize total costs, and the lower level represents evasive drivers selecting routes to avoid facilities. Dan and Marcotte (2019) studied a competitive bilevel problem that integrates facility competition, congestion, and stochastic user behavior. The upper level determines the location to maximize market share. The lower-level models selfish users choosing facilities that maximize their own utility, factoring in travel time, queuing delays, and random preferences. Sugishita et al. (2025) studied a bilevel network design problem, and their primary focus is on expanding the capacity of stations that already exist. The distinction between their work and ours lies in the modeling of queuing effects. In their study, queuing time is represented implicitly through classical arc travel-time functions. In contrast, our work explicitly models congestion-induced queuing delays at charging facilities using an $M/M/c$ system, where charging demand is endogenously determined through EV path selection.

To the best of our knowledge, the following two bilevel studies are the most closely related to this work. In Huang and Kockelman (2020)'s research, the upper level determines the locations to maximize the profits of the station owners. The lower level represents user equilibrium, where EV drivers select routes and stations based on travel time, charging time, and queuing time. However, their modeling of queuing time is based on the assumption that queues occur when the demand at a charging station exceeds 80% of its charging capacity. Kınay, Gzara, and Alumur (2023) considered a bilevel framework for simultaneously determining station locations and capacities under stochastic demand and congestion. They adopted $M/M/c$ queuing theory to model congestion at the station. However, they only require that the probability of an EV's queuing time at a station not exceeding a given threshold be greater than a prescribed level, and the resulting queuing time is not incorporated into EV path selection. In other words, they assumed that EVs always select the shortest-distance path and congestion is used only to evaluate the feasibility of station capacity, not to influence route selection.

In summary, although prior studies have considered station capacity, stochastic demand, discrete choice behavior, or bilevel responses, none of them jointly model EV path choice and congestion-induced queuing delays within a unified bilevel location framework. Existing bilevel models either simplify queuing effects or ignore their impact on path selection. No existing work captures the feedback between route decisions and charging facility congestion. Importantly, ignoring queuing time in EV path selection may lead to unrealistic routing predictions: the shortest-distance path can incur substantial waiting delays at charging stations, causing the total trip duration to exceed drivers' tolerance. In such cases, EV drivers may instead choose alternative routes that involve a small detour but offer shorter queues and a lower overall travel time. This gap motivates the bilevel modeling and algorithmic framework developed in this paper.

3 Problem description

This section introduces the bilevel charging facility location problem (BCLP), which explores how the expected queuing time and the number of charging stops along a path influence EVs' travel behaviors.

3.1 Leader and follower levels

We consider a transportation network $G = (N, A)$ with a set of transportation arcs A, a set of candidate nodes N, a set of deployment configurations W, and a set of EV demands K. Each configuration $j \in W$ includes the number of chargers n_j to be deployed, and the corresponding deployment cost c_j. Each candidate node $i \in N$ can represent a station, area, city, or population center where a configuration can

be deployed. Each demand $k \in K$ is associated with a stochastic stream of EV flow f_k between an OD pair (O_k, D_k), an EV driving range R_k, and a maximum tolerable trip duration threshold λ_k, which accounts for travel time, charging time, and expected queuing time. With this definition, multiple EV demands associated with the same OD pair, characterized by different flow volumes, driving ranges, and deviation tolerances, can be considered.

At the leader level, one needs to determine the location decisions and the corresponding deployment configurations. The objective of the leader level is to maximize the total covered flow under a given investment budget B. At the follower level, EV drivers respond to the location decisions from the leader level. We design a disutility function that accounts for expected queuing time and charging stops to model their path choice behavior. In line with the literature, we make the following assumptions. **(A1)**: The driving speed of the EVs and the charging time per stop are constant. **(A2)**: To mitigate range anxiety and ensure a satisfactory level of service from the charging provider's perspective, EVs are assumed to depart with half of their battery capacity unless a charging facility is open at the origin. They must arrive at the destination with at least half of their battery remaining unless a charging facility is open at the destination. This assumption has been widely adopted in prior studies (Arslan et al. 2019, Kınay, Gzara, and Alumur 2021, Mahmutoğulları and Yaman 2024), and can be easily relaxed through a standard graph construction in a preprocessing step. **(A3)**: For each demand $k \in K$, the associated EVs are assumed to select at most one charging path for their trip or choose to opt out (Xu and Meng 2020, Park and Lee 2024). This assumption can be easily relaxed by defining multiple homogeneous demands on the same OD pair.

The EV drivers typically prefer paths with shorter distances, shorter queuing delays, and fewer charging stops (Xie et al. 2018, Kınay, Gzara, and Alumur 2023, Park and Lee 2024). We design the disutility function to incorporate these three key components, enabling us to evaluate every possible path. Let P_k be the set of all possible charging paths for demand k. For each path $p \in P_k$, d_p represents the travel distance of path p, v is the travel speed, ξ represents the constant charging time per stop, N_p denotes the set of charging nodes visited along path p, and q_i represents the expected queuing time at each visited charging node. We define the total time variable u_p as the disutility of each path as follows:

$$u_p = \frac{d_p}{v} + \xi|N_p| + \sum_{i \in N_p} q_i \qquad \forall k \in K, p \in P_k, \tag{1}$$

where u_p includes the travel time on the road, the total charging time, and the total expected queuing time.

3.2 Modeling the expected queuing time

The expected queuing time at a charging node is jointly determined by the deployment configuration and the EV flow passing through it. Similar to Xie et al. (2018) and Kınay, Gzara, and Alumur (2023), we consider an $M/M/c$ queuing system to capture congestion and estimate the expected queuing time.

We assume that the stochastic EV flow rate of each demand follows a Poisson distribution. We define the decision variable Δ_i as the sum of the EV flow rates of all demands passing through the charging node i, determined as the total Poisson OD demand. Let e be the exponential service rate of each charger. To maintain consistency of path disutility (1), we approximate the constant charging time by a deterministic value $\xi = 1/e$ per charging stop. If we deploy configuration $j \in W$ at the node, n_j represents the number of charging facilities deployed. Thus, we need the condition $\Delta_i < en_j$ to maintain stability; that is, the arrival rate must remain below the total service capacity; otherwise, the queuing length will be infinite. The average charging load at node i is $\eta = \frac{\Delta_i}{en_j}$. We use a modification of the classical Erlang-C formula (Chromy, Misuth, and Weber 2012, Xie et al. 2018) to calculate the

probability that a vehicle needs to queue for charging at node i under deployment configuration j:

$$\mathbb{P}(\Delta_i, j) = \frac{\frac{(n_j\eta)^{n_j}}{n_j!(1-\eta)}}{\frac{(n_j\eta)^{n_j}}{n_j!(1-\eta)} + \sum_{t=0}^{n_j-1} \frac{(n_j\eta)^t}{t!}}.$$

The corresponding expected queuing time φ_{ij} at node i under deployment configuration j is computed accordingly using the same modified Erlang-C formulation (Chromy, Misuth, and Kavacky 2011, Xie et al. 2018):

$$\varphi_{ij} = \frac{\mathbb{P}(\Delta_i, j)}{en_j - \Delta_i} = \frac{\frac{(n_j\eta)^{n_j}}{n_j!(1-\eta)}}{\frac{(n_j\eta)^{n_j}}{n_j!(1-\eta)} + \sum_{t=0}^{n_j-1} \frac{(n_j\eta)^t}{t!}} \frac{1}{en_j - \Delta_i}. \tag{2}$$

Fig. 1 shows an example of the modeled $M/M/c$ queuing function (yellow line) that illustrates the relationship between charging demand (EV flow rate) Δ and expected queuing time φ at node i. When the charging demand is low, the expected queuing time is nearly zero and increases slowly. As the charging demand approaches the charging capacity, the queuing time rises sharply and tends toward infinity. To facilitate modeling, we approximate the $M/M/c$ queuing function using a piecewise-linear (PWL) function (black dashed line) and define a maximum queuing time, truncating the function curve at q_{max}. Note that different deployment configurations have different queuing functions.

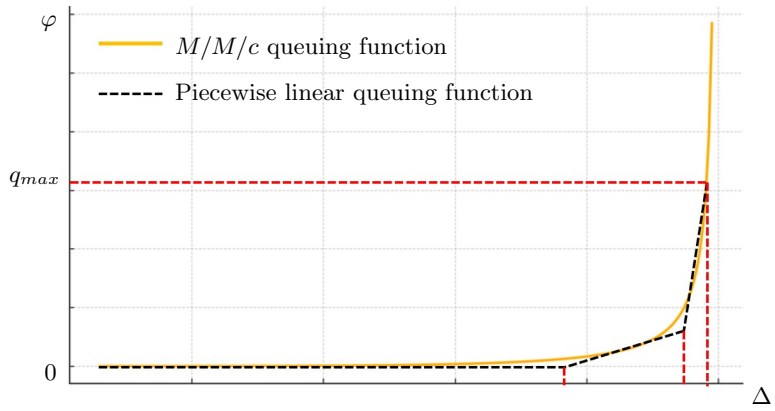


Figure 1: $M/M/c$ **Queuing model: EV flow rate and expected queuing time**

We define I_j as the set of breakpoints of the PWL function corresponding to each deployment configuration $j \in W$. For each breakpoint $l \in I_j$, we define (U_j^l, J_j^l) as the queuing time and EV flow rate corresponding to that point in the PWL function. Since we truncate the curve below $q_{\max}$ in the PWL approximation, for the last breakpoint, $U_j^{|I_j|} = q_{\max}$ and $J_j^{|I_j|}$ denotes the maximum EV flow rate that deployment configuration j can handle. Note that the maximum flow value of the last breakpoint is slightly below en_j ($J_j^{|I_j|} < en_j$). We define the parameter $\rho_j^l = \frac{U_j^l - U_j^{l-1}}{J_j^l - J_j^{l-1}}$ as the slope of the l-th segment and define the parameter $\mu_j^l = J_j^l - J_j^{l-1}$ be the length of the EV flow rate in l-th segment. The slope of the first segment is set to 0, while the slopes of subsequent segments strictly increase, i.e., $\rho_j^l < \rho_j^{l+1}$. The queuing time φ_{ij} at node i under configuration j after the piecewise-linear approximation is given as follows:

$$\varphi_{ij} = \begin{cases} 0 & \text{if } 0 \le \Delta_i \le J_j^1 \\ \rho_j^2(\Delta_i - J_j^1) & \text{if } J_j^1 \le \Delta_i \le J_j^2 \\ U_j^2 + \rho_j^3(\Delta_i - J_j^2) & \text{if } J_j^2 \le \Delta_i \le J_j^3 \\ \cdots & \cdots \\ U_j^{|I_j|-1} + \rho_j^{|I_j|}(\Delta_i - J_j^{|I_j|-1}) & \text{if } J_j^{|I_j|-1} \le \Delta_i \le J_j^{|I_j|}. \end{cases}$$

To model the PWL function, we introduce the following decision variables: continuous variable q_i representing the expected queuing time at node i; binary decision variable w_{ij} indicating whether deployment configuration j is installed at node i; binary decision variable σ_{ij}^l indicating whether the l-th segment for deployment configuration j at node i is activated; and continuous variable $b_{ij}^l \in [0,1]$ representing the proportion of activation of the l-th segment for deployment configuration j at node i. The PWL function can be formulated as follows:

$$\Delta_i \leq \sum_{j \in W} w_{ij} J_j^{|I_j|} \qquad \forall i \in N \tag{3}$$

$$\Delta_i = \sum_{j \in W} \sum_{l \in I_j} \mu_j^l b_{ij}^l \qquad \forall i \in N \tag{4}$$

$$q_i = \sum_{j \in W} \varphi_{ij} = \sum_{j \in W} \sum_{l \in I_j} \rho_j^l \mu_j^l b_{ij}^l \qquad \forall i \in N \tag{5}$$

$$\sigma_{ij}^{|I_j|} \leq \sigma_{ij}^{|I_j|-1} \leq \ldots \leq \sigma_{ij}^1 \leq w_{ij} \qquad \forall i \in N, j \in W \tag{6}$$

$$0 \leq \sigma_{ij}^{l+1} \leq b_{ij}^l \leq \sigma_{ij}^l, \; 0 \leq b_{ij}^{|I_j|} \leq \sigma_{ij}^{|I_j|} \qquad \forall i \in N, j \in W, l \in \{1, \ldots, |I_j|-1\} \tag{7}$$

$$w_{ij} \in \{0,1\}, \; \sigma_{ij}^l \in \{0,1\}, \; 0 \leq b_{ij}^l \leq 1, \; \Delta_i \geq 0, \; q_i \geq 0 \qquad \forall i \in N, j \in W, l \in I_j. \tag{8}$$

Each node can deploy at most one configuration. Therefore, constraints (3) ensure that the EV flow that passes through a node does not exceed the maximum flow that the installed deployment configuration can handle. Constraints (4) track the EV flow rate in the PWL function, which is equal to the sum of the lengths of all activated segments for the installed deployment configuration. Constraints (5) track the queuing time in the PWL function, which is the sum of the queuing times associated with the activated segments of the installed deployment configuration. Constraints (6) represent the active sequence order of σ_{ij}^l. The binary variables σ_{ij}^l enforce the activation order of the incremental PWL segments, ensuring that the segment cannot receive any flow unless all previous segments are entirely used. If a deployment configuration is not installed at a node, then all associated segment-activation variables are set to zero. Constraints (7) ensure that when a segment is activated, the corresponding proportion decision variable in the preceding segment is forced to take the value 1. Constraints (8) define the domain of decision variables.

3.3 Path selection of EV drivers

The path selection logic of EV drivers plays a key role in the interaction between charging facility deployment and congestion effects. We describe the path selection logic as follows.

3.3.1 Feasibility and time-threshold conditions

For a given location decision at the leader level, EV drivers associated with each demand $k \in K$ may choose a charging path or opt out of travel. A charging path is considered feasible if it satisfies the following conditions:

(i) Charging Facility Availability. All charging nodes along the selected path must be open to ensure that the path is feasible, considering the ranges of the EVs.

(ii) Time Deviation-Boundedness. Similar to prior studies (Yıldız, Arslan, and Karaşan 2016, Arslan et al. 2019, Kınay, Gzara, and Alumur 2021, Mahmutoğulları and Yaman 2023) that allow drivers to deviate from the shortest-distance path, we consider that the total duration of the selected path cannot exceed the shortest-trip-time by more than a prescribed threshold. For each demand k, the total time of the selected path $p \in P_k$ (sum of travel, charging, and queuing time) does not exceed a threshold λ_k, that is, $u_p \leq \lambda_k$. We set the parameter $\lambda_k = (1+\tau_k)\lambda_k^{min}$, where λ_k^{min} is the time required to traverse the shortest-distance path with the fewest charging stops and without queuing delays. The construction of λ_k uses a no-queue baseline λ_k^{min} and a time deviation tolerance factor τ_k.

3.3.2 ϵ-optimality

From the EV drivers' perspective, a natural and "selfish" setting would be that they always select the optimal feasible charging path with the minimum disutility. However, the disutility function (1) of a charging path includes queuing time, which is endogenously influenced by EV path selections. As shown in Figure 2, EVs' path selections affect the charging demand at each charging node, thereby impacting queuing delays, which in turn alter path disutilities. This feedback mechanism implies that forcing EVs to select an optimal path with the minimum disutility may, for certain location decisions, lead to the absence of a stable and feasible response.

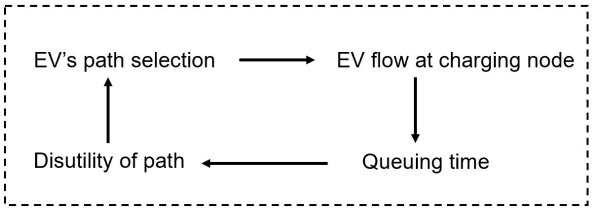


Figure 2: Endogenous feedback loop between path selection and queuing delays

To illustrate this issue, consider the example shown in Figure 3. Suppose that only a single demand k with EV flow f_k is considered and that two symmetric charging paths are available, each passing through a distinct charging facility, namely station A and station B. The two stations are assumed to have identical queuing functions. In particular, when the EV flow f_k passes through a station, the expected queuing time is one minute.

When both station A and station B are opened simultaneously, path 1 and path 2 are both feasible. In this case, no feasible path selection for EV drivers exists under strict optimality. Indeed, if EV drivers choose path 1, they incur a queuing time of one minute at station A, while path 2 remains feasible with zero queuing time and thus a lower disutility. Conversely, if path 2 is chosen, path 1 becomes strictly better. As a result, EV drivers cannot select any path that minimizes disutility, and no strictly optimal path selection exists. This phenomenon reflects a general challenge arising from discrete path selections combined with congestion-dependent costs: strict best-response dynamics do not necessarily admit a fixed point for all leader decisions.

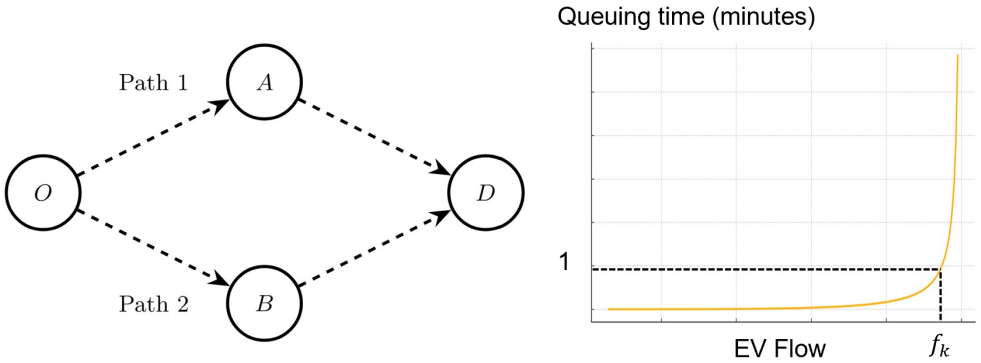


Figure 3: An illustrative example of feedback loop between path selection and queuing delays

To ensure that each location decision can be consistently evaluated at the follower level, we adopted the concept of ϵ-optimality (Besançon, Anjos, and Brotcorne 2021, 2024). Rather than requiring EV drivers to select a path with minimum disutility, we allow them to choose any feasible path whose disutility is within ϵ of the minimum achievable value. Formally, a path $p \in P_k$ is admissible and can be selected if $u_p \leq u_k^* + \epsilon$, where u_k^* denotes the minimum disutility among all feasible charging

paths for demand k. Under the setting of ϵ-optimality, EV drivers are allowed to select any feasible suboptimal paths within ϵ range. In the example shown in Figure 3, when ϵ is set to one minute, both path 1 and path 2 become admissible charging paths when stations A and B are simultaneously opened.

The ϵ-optimality condition serves two purposes. First, it guarantees the existence of a stable follower response for the leader's decisions, even in the presence of congestion-induced feedback loops. Second, it provides a realistic representation of bounded rationality and imperfect information, under which drivers may tolerate small deviations from strict optimality. Importantly, ϵ-optimality should be interpreted as a modeling device that stabilizes the follower's response rather than as an exact behavioral rule.

3.3.3 Cooperative (optimistic) response and planning perspective

In our bilevel problem, multiple follower responses within ϵ-optimality may exist for a given location decision. We adopt a cooperative (optimistic) response assumption (Kınay, Gzara, and Alumur 2023), under which the follower's solution selected is one that maximizes the leader's covered flow among all admissible ϵ-optimal responses.

This cooperative assumption should not be interpreted as a literal description of decentralized driver behavior. Instead, it provides a planning-oriented upper bound on the achievable coverage under the given infrastructure deployment, corresponding to scenarios in which some degree of coordination, guidance, or information provision (e.g., routing recommendations or reservation systems) is available (Guillet and Schiffer 2025). As such, the cooperative response allows the leader to evaluate charging facility deployments under favorable but well-defined user reactions, ensuring consistency and tractability of the bilevel model.

4 Mathematical models

This section introduces two bilevel mathematical models for solving the BCLP: an arc-flow model and a path-based model. For the reader's convenience, Tables A1 and A2 in the Appendix summarize the sets, parameters, and variables used in the model.

4.1 Bilevel arc-flow based model

We adopt the network transformation notion (MirHassani and Ebrazi 2013, Yıldız, Arslan, and Karaşan 2016, Arslan et al. 2019, Kınay, Gzara, and Alumur 2021) and expand the network of each demand $k \in K$ to a transformed network $G_k(N_k, A_k)$. We note that the existing network transformation approaches define the node set N_k based on a distance threshold. In contrast, our problem formulation is based on a time-threshold λ_k, which limits the maximum tolerable trip duration. To align with the transformed-network construction, we exploit Assumption (A1), under which the travel speed v is constant. According to the time threshold λ_k, the sum of the distances from node i to s_k and t_k cannot exceed $v\lambda_k$. Let δ_{ij} be the shortest path distance in G between i to j, let (s_k, t_k) be the dummy origin node and destination node of demand k. We define the node set $N_k = \{s_k, t_k\} \cup \{i \in N : \delta_{O_k i} + \delta_{i D_k} \leq v\lambda_k\}$, which represents the set of nodes whose distances fall within $v\lambda_k$. The arc set $A_k = A_k^1 \cup A_k^2 \cup A_k^3$, where:

$$
\begin{aligned}
A_k^1 &= \{(s_k, j) : \delta_{O_k j} \leq R_k/2, j \in N_k \backslash \{s_k, t_k\}\}, \\
A_k^2 &= \{(i, t_k) : \delta_{i D_k} \leq R_k/2, i \in N_k \backslash \{s_k, t_k\}\}, \\
A_k^3 &= \{(i, j) : \delta_{ij} \leq R_k, i, j \in N_k \backslash \{s_k, t_k\}, i \neq j\}.
\end{aligned}
$$

To model the bilevel problem, we introduce the following decision variables:

$$x_{ijk} = \begin{cases} 1 & \text{if arc } (i,j) \text{ in the expanded network is on the selected (realized) path of demand } k \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{x}_{ijk} = \begin{cases} 1 & \text{if arc } (i,j) \text{ in the expanded network is on the minimum-disutility reference path for demand } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if a charging facility is open at node } i \\ 0 & \text{otherwise} \end{cases}$$

$$w_{ij} = \begin{cases} 1 & \text{if deployment configuration } j \text{ is installed at node } i \\ 0 & \text{otherwise} \end{cases}$$

$$z_k = \begin{cases} 1 & \text{if demand } k \text{ is satisfied} \\ 0 & \text{otherwise.} \end{cases}$$

We use two arc-indicator variable types to distinguish the realized charging path from a minimum-disutility benchmark path. The binary variable x_{ijk} indicates whether arc (i,j) of demand k is on the selected (realized) charging path, i.e., the path ultimately taken by the traveler under the leader's design. In contrast, $\hat{x}_{ijk}$ denotes an optimal-reference charging path for demand k: it represents a feasible path that attains the minimum disutility for the same leader-induced conditions (open facilities and the resulting queuing times). When multiple minimum-disutility paths exist, $\hat{x}_{ijk}$ may correspond to any one of them.

The bilevel arc-flow model [BLM-A] is formulated as follows:

$$\max \quad \sum_{k\in K} f_k z_k \tag{9}$$

s.t. PWL queuing function: (3) – (8)

$$\sum_{j\in W} w_{ij} = y_i \qquad \forall i \in N \tag{10}$$

$$\sum_{i\in N}\sum_{j\in W} c_j w_{ij} \le B \tag{11}$$

$$\Delta_i = \sum_{k\in K}\sum_{(i,j)\in A_k, i\neq s_k} f_k x_{ijk} \qquad \forall i \in N \tag{12}$$

$$\sum_{j:(i,j)\in A_k} x_{ijk} - \sum_{j:(j,i)\in A_k} x_{jik} = \begin{cases} z_k, & \text{if } i = s_k \\ -z_k, & \text{if } i = t_k \\ 0, & \text{otherwise} \end{cases} \qquad \forall k \in K, i \in N_k \tag{13}$$

$$\sum_{i:(i,j)\in A_k} x_{ijk} \le y_j \qquad \forall k \in K,\ j \in N_k, j \neq t_k \tag{14}$$

$$\sum_{(i,j)\in A_k} \frac{\delta_{ij}}{v} x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \xi x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} q_j x_{ijk} - \left(\sum_{(i,j)\in A_k} \frac{\delta_{ij}}{v} \hat{x}_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \xi \hat{x}_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} q_j \hat{x}_{ijk} \right) \le z_k \epsilon \qquad \forall k \in K \tag{15}$$

$$\sum_{(i,j)\in A_k} \frac{\delta_{ij}}{v} x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \xi x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} q_j x_{ijk} - \lambda_k z_k \le 0 \qquad \forall k \in K \tag{16}$$

$$x_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A_k \tag{17}$$

$$y_i \in \{0,1\} \qquad \forall i \in N \tag{18}$$

$$(\hat{x}_{ijk}, z_k) \in \operatorname{argmin} \sum_{(i,j)\in A_k} \frac{\delta_{ij}}{v}\hat{x}_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \xi\hat{x}_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} q_j\hat{x}_{ijk} - \lambda_k z_k \quad \forall k \in K \tag{19}$$

$$s.t. \sum_{j:(i,j)\in A_k} \hat{x}_{ijk} - \sum_{j:(j,i)\in A_k} \hat{x}_{jik} = \begin{cases} z_k, & \text{if } i = s_k \\ -z_k, & \text{if } i = t_k \\ 0, & \text{otherwise} \end{cases} \quad \forall k \in K, i \in N_k \tag{20}$$

$$\sum_{i:(i,j)\in A_k} \hat{x}_{ijk} \leq y_j \quad \forall k \in K, j \neq t_k \tag{21}$$

$$z_k \in \{0,1\} \quad \forall k \in K \tag{22}$$

$$\hat{x}_{ijk} \in \{0,1\} \quad \forall k \in K, (i,j) \in A_k. \tag{23}$$

The objective function (9) of the leader is to maximize the EV flow. The PWL queuing functions between q_i and Δ_i are already introduced in constraints (3) to (8). Constraints (10) ensure that the deployment configuration can only be installed if a charging facility is open at the node. Constraint (11) states that the total cost of charging facilities and the deployment configuration must not exceed the budget. Constraints (12) track the charging demand at each node. Constraints (13) represent the flow balance of the selected path for each demand. Although x_{ijk} appears in upper-level constraints (because it drives facility loads/queuing and coverage), it should be interpreted as a follower response variable determined by the traveler's path choice given (y, w) through the induced q; $\hat{x}_{ijk}$ is used only as a benchmark to define and enforce the ϵ-optimality (bounded rationality) condition. Constraints (14) ensure that the nodes along the selected path must be open. Constraints (15) represent ϵ-optimality, meaning that for each demand, the disutility of the selected path cannot exceed that of the optimal path with minimum disutility by more than ϵ. Constraints (16) state that, for each demand k, the disutility of the selected path does not exceed λ_k. Constraints (17) to (18) define the domain of the decision variables.

In the follower's problem, we need to identify the optimal path for each demand and its corresponding disutility. We use constraints (19) to represent the objective function of the follower, and constraints (20) to represent the flow balance of the optimal path. The follower's objective function and the flow balance can be interpreted as evaluating: $\min\left\{0, \min_{p\in P_k}\{u_p - \lambda_k z_k\}\right\}$. If there exists a feasible minimum-disutility path p with $u_p \leq \lambda_k$, choosing that path yields a negative objective value and is therefore optimal; in this case, the demand is satisfied and $z_k = 1$. Otherwise, all feasible paths satisfy $u_p > \lambda_k$; opting out ($z_k = 0$) becomes optimal because it yields objective value 0, which is no worse than taking any feasible path. Constraints (21) state that the nodes along the selected path must be open.

4.2 Bilevel path-based model

The bilevel path-based model and the bilevel arc-flow model are the same in terms of tracking charging demand, queuing function, and deployment configuration. The difference is that we construct the follower-level model by considering the set of all possible charging paths P_k for each demand k within the time deviation range. To calculate the total time (disutility) of path p, we define parameter t_p as the sum of the travel time and charging time along the path p without queues, i.e., $t_p = d_p/v + \xi|N_p|$. The disutility of each path is $u_p = t_p + \sum_{i\in N_p} q_i$. We drop the decision variables x_{ijk} and $\hat{x}_{ijk}$ in BLM-A and introduce the new decision variables:

$$x_p = \begin{cases} 1 & \text{if charging path } p \in P_k \text{ is the selected (realized) path} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{x}_p = \begin{cases} 1 & \text{if charging path } p \in P_k \text{ is the minimum-disutility reference path} \\ 0 & \text{otherwise} \end{cases}$$

together with the new binary parameters a_{pi} equal to 1 if the selected path $p \in P_k$ visits charging node $i \in N$ and 0 otherwise. The bilevel path-based model [BLM-P] is formulated as follows:

$$
\begin{aligned}
\max \quad & (9) \\
\text{s.t.} \quad & (3)-(8),(10)-(11) \\
& \Delta_i = \sum_{k\in K}\sum_{p\in P_k} f_k x_p a_{pi} && \forall i \in N && (24)\\
& \sum_{p\in P_k} x_p - z_k = 0 && \forall k \in K && (25)\\
& x_p \le y_i && \forall k \in K, p \in P_k, i \in N_p && (26)\\
& \sum_{p\in P_k} t_p x_p + \sum_{p\in P_k} x_p \sum_{i\in N_p} q_i - \left(\sum_{p\in P_k} t_p \hat{x}_p + \sum_{p\in P_k} \hat{x}_p \sum_{i\in N_p} q_i \right) \le z_k \epsilon && \forall k \in K && (27)\\
& \sum_{p\in P_k} t_p x_p + \sum_{p\in P_k} x_p \sum_{i\in N_p} q_i \le \lambda_k z_k && \forall k \in K && (28)\\
& z_k \in \{0,1\} && \forall k \in K && (29)\\
& x_p \in \{0,1\} && \forall k \in K, p \in P_k && (30)\\
& (\hat{x}_p, z_k) \in \operatorname{argmin} \sum_{p\in P_k} t_p \hat{x}_p + \sum_{p\in P_k} \hat{x}_p \sum_{i\in N_p} q_i - \lambda_k z_k && \forall k \in K && (31)\\
& \quad s.t. \sum_{p\in P_k} \hat{x}_p - z_k = 0 && \forall k \in K && (32)\\
& \hat{x}_p \le y_i && \forall k \in K, p \in P_k, i \in N_p && (33)\\
& \hat{x}_p \in \{0,1\} && \forall k \in K, p \in P_k. && (34)
\end{aligned}
$$

Constraints (24) track the charging demand at each charging node. Constraints (25) ensure that when a demand is satisfied, the driver can only select one path to travel. Constraints (26) ensure that the nodes along the selected path must be open. Constraints (27) ensure the ϵ-optimality for each demand, meaning that the difference between the disutility of the selected path and that of the optimal path is within ϵ. Constraints (28) ensure the disutility of the selected path cannot exceed the threshold λ_k. Constraints (29) to (30) define the domain of the decision variables. The objective function (31) of the follower is to identify the optimal path with the minimum disutility within the threshold λ_k for each demand k. Constraints (32) ensure that when a demand is satisfied, the optimal path exists. Constraints (33) ensure that the charging nodes along the optimal path must be open.

5 Reformulations to single-level models

To solve the bilevel models BLM-A and BLM-P, we relax, without loss of generality, the integrality constraints in the follower's problem and reformulate them as single-level MILP models using strong duality.

Proposition 1. *The follower's problem in models BLM-A and BLM-P with relaxed integrality constraints on x and z variables always has an integer optimal solution when feasible.*

Proof. We sketch the argument for the BLM-P model; the proof for BLM-A is analogous.

For each demand k, consider adding a dummy path p_0^k whose total time equals the threshold $u_{p_0^k} = \lambda_k$ and imposing the constraint $\sum_{p\in P_k\cup\{p_0^k\}} x_p = 1$. Selecting p_0^k corresponds to opting out, i.e., the demand k is not satisfied. In fact, $z_k = 1 - x_{p_0^k}$. After relaxing the integrality constraints to $0 \le x_p \le 1$, the follower's problem for demand k can be written as the linear program

$$
\min \left\{ \sum_{p\in P_k\cup\{p_0^k\}} u_p x_p : \sum_{p\in P_k\cup\{p_0^k\}} x_p = 1,\ 0 \le x_p \le 1 \right\},
$$

which ensures that we must identify the optimal path with the minimum total time from the set $P_k \cup p_0^k$. When path p_0^k is identified, it means that the demand k can not be satisfied.

Suppose that a fractional optimal solution exists. Without loss of generality, we assume that two paths p_1 and p_2 have the corresponding fractional flows x_{p_1} and x_{p_2}, and that u_{p_1} and u_{p_2} denote their corresponding disutilities. We have $x_{p_1} + x_{p_2} = 1$. Due to the optimality of the solution, it must hold that $u_{p_1} = u_{p_2}$; otherwise, one could select only the path $p^* = \arg\min\{u_{p_1}, u_{p_2}\}$ and obtain a lower objective value because $\min\{u_{p_1}, u_{p_2}\} \leq x_{p_1} u_{p_1} + x_{p_2} u_{p_2}$. Since $u_{p_1} = u_{p_2}$, we can arbitrarily choose one of the two paths, set its corresponding decision variable x_p to 1, and set $x_p = 0$ for all remaining paths. Therefore, the follower's problem of the BLM-P model exhibits a zero optimality gap property, meaning that the optimal solution value of the relaxed model is equal to that of the integer model. □

5.1 Reformulation of model BLM-A

We relax the follower's domain restriction constraints on variables z_k into $0 \leq z_k \leq 1$. Let π_{ik}, γ_{ik}, and θ_k be the dual variables associated with follower's constraints (20), (21), and the constraints $z_k \leq 1$ due to (22), respectively. The dual model of the relaxed follower's problem in BLM-A is as follows:

$$\max \sum_{i \in N_k, i \neq t_k} \gamma_{ik} y_i + \theta_k \qquad \forall k \in K \tag{35}$$

$$\text{s.t. } \pi_{ik} - \pi_{jk} + \gamma_{jk} \leq \frac{\delta_{ij}}{v} + \xi + q_j \qquad \forall k \in K, (i,j) \in A_k, j \neq t_k \tag{36}$$

$$\pi_{ik} - \pi_{t_k,k} \leq \frac{\delta_{i,t_k}}{v} \qquad \forall k \in K, (i,t_k) \in A_k \tag{37}$$

$$\pi_{s_k,k} - \pi_{t_k,k} - \theta_k \geq \lambda_k \qquad \forall k \in K \tag{38}$$

$$\pi_{ik} \text{ is free} \qquad \forall k \in K, (i,j) \in A_k \tag{39}$$

$$\theta_k \leq 0 \qquad \forall k \in K \tag{40}$$

$$\gamma_{ik} \leq 0 \qquad \forall k \in K, i \in N_k, i \neq t_k. \tag{41}$$

The constraints (36)–(41) ensure the dual feasibility of the follower problem, and the strong duality condition for the follower is given as follows:

$$\sum_{(i,j) \in A_k} \frac{\delta_{ij}}{v} \hat{x}_{ijk} + \sum_{(i,j) \in A_k, j \neq t_k} \xi \hat{x}_{ijk} + \sum_{(i,j) \in A_k, j \neq t_k} q_j \hat{x}_{ijk} - \lambda_k z_k = \sum_{i \in N_k, i \neq t_k} \gamma_{ik} y_i + \theta_k \quad \forall k \in K. \tag{42}$$

Therefore, we introduce the following lemma.

Lemma 1. *[ϵ-optimality through strong duality.] Consider the following condition (43):*

$$\left(\sum_{(i,j) \in A_k} \frac{\delta_{ij}}{v} x_{ijk} + \sum_{(i,j) \in A_k, j \neq t_k} \xi x_{ijk} + \sum_{(i,j) \in A_k, j \neq t_k} q_j x_{ijk} - \lambda_k z_k \right) - \left(\sum_{i \in N_k, i \neq t_k} \gamma_{ik} y_i + \theta_k \right) \leq z_k \epsilon \forall k \in K \tag{43}$$

together with dual feasibility conditions (36) to (41). Any feasible primal–dual pair satisfying the above inequality guarantees that the disutility of the selected path is ϵ-optimal with respect to the true optimal value of the follower's problem.

Proof. By weak duality, for any primal–dual feasible pair, the primal objective value is greater than or equal to the dual objective value. Therefore, the relationship between the disutility of the selected path, the disutility of the optimal path, and the dual objective value of the follower is shown in

Figure 4. We do not need to explicitly know the disutility of the optimal path; instead, we can use its dual value as a lower bound and impose constraints (43) to ensure that the gap between the disutility of the selected path and this lower bound remains within ϵ. When $z_k = 0$, both x_{ijk} and $\hat{x}_{ijk}$ are equal to zero. When $z_k = 1$, the above inequality implies that the selected primal solution is at most ϵ away from the true optimum. When $\epsilon = 0$, the selected path is the optimal path with minimum disutility. Hence, the corresponding selected path is ϵ-optimal. □

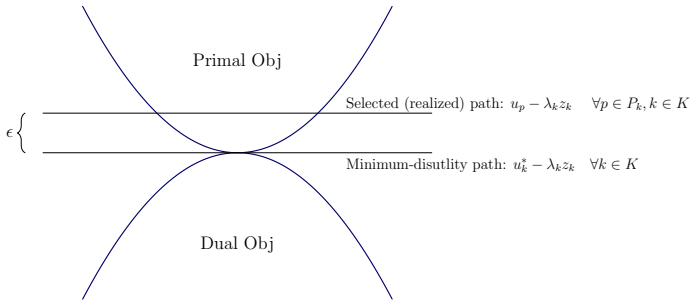


Figure 4: Illustration of the disutility gap between the selected path and the minimum-disutility path

After introducing Lemma 1, we can replace constraints (15) in BLM-A with constraints (43) and add the dual feasibility conditions (36)–(41). The constraints (19)–(23) related to $\hat{x}_{ijk}$ can then be removed.

Note that condition (43) involves two non-linear terms, and constraints (16) in BLM-A involve one non-linear term, which requires linearization. First, we introduce the auxiliary variables $\hat{q}_{ijk} = q_j x_{ijk}$ and $\hat{\gamma}_{ik} = \gamma_{ik} y_i$. Second, we introduce the parameter $q_{\max}$ as an upper bound on q_j, and the parameter $-\lambda_k$ as a lower bound on γ_{ik}. Since the follower objective value is always larger than $-\lambda_k$, the shadow price of opening a node cannot improve the follower objective by more than λ_k, hence, $\gamma_{ik} \geq -\lambda_k$ holds without loss of optimality.

Therefore, condition (43) can be replaced by (44), and constraints (16) in BLM-A can be replaced by (45),

$$\left(\sum_{(i,j)\in A_k} \frac{\delta_{ij}}{v} x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \xi x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \hat{q}_{ijk} - \lambda_k z_k \right) - \left(\sum_{i\in N_k, i\neq t_k} \hat{\gamma}_{ik} + \theta_k \right) \leq z_k \epsilon \qquad \forall k \in K \qquad (44)$$

$$\sum_{(i,j)\in A_k} \frac{\delta_{ij}}{v} x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \xi x_{ijk} + \sum_{(i,j)\in A_k, j\neq t_k} \hat{q}_{ijk} - \lambda_k z_k \leq 0 \qquad \forall k \in K \qquad (45)$$

and linearized through constraints (46)–(50) as follows:

$$\gamma_{ik} \leq 0 \qquad \forall k \in K, i \in N_k, i \neq t_k \qquad (46)$$

$$0 \leq \hat{q}_{ijk} \leq q_{max} x_{ijk} \qquad \forall k \in K, (i,j) \in A_k, j \neq t_k \qquad (47)$$

$$q_j + q_{max}(x_{ijk} - 1) \leq \hat{q}_{ijk} \leq q_j \qquad \forall k \in K, (i,j) \in A_k, j \neq t_k \qquad (48)$$

$$-\lambda_k y_i \leq \hat{\gamma}_{ik} \leq 0 \qquad \forall k \in K, i \in N_k / \{t_k\} \qquad (49)$$

$$\gamma_{ik} \leq \hat{\gamma}_{ik} \leq \gamma_{ik} + (1 - y_i)\lambda_k \qquad \forall k \in K, i \in N_k / \{t_k\}. \qquad (50)$$

Finally, we can reformulate the BLM-A into the single-level mixed-integer linear programming model [SLM-A].

[SLM-A] max (9)
s.t. Constraints (3)–(8): PWL queuing function
Constraints (10)–(12): Budget and flow capacity
Constraints (13)–(18): Follower's condition for the selected path
Constraints (44)–(45): Follower's condition for ϵ-optimality
Constraints (36)–(41): Follower's dual feasibility condition
Constraints (46)–(50): Linearization constraints.

5.2 Reformulation of model BLM-P

We apply a similar process to the BLM-P model. Let π_k, γ_{kpi} and θ_k be the dual variables associated with Constraints (32), (33) and (29), respectively. The strong duality condition of the follower's problem is as follows:

$$\sum_{p\in P_k} t_p x_p + \underbrace{\sum_{p\in P_k} x_p \sum_{i\in N_p} q_i}_{\text{Requires linearization}} - \lambda_k z_k = \underbrace{\sum_{p\in P_k, i\in N_p} y_i \gamma_{kpi}}_{\text{Requires linearization}} + \theta_k \qquad \forall k \in K. \tag{51}$$

The dual feasibility requirements are as follows:

$$\pi_k + \sum_{i\in N_p} \gamma_{kpi} \le t_p + \sum_{i\in N_p} q_i \qquad \forall k \in K, p \in P_k \tag{52}$$
$$\pi_k - \theta_k \ge \lambda_k \qquad \forall k \in K \tag{53}$$
$$\pi_k \text{ is free} \qquad \forall k \in K \tag{54}$$
$$\theta_k \le 0 \qquad \forall k \in K \tag{55}$$
$$\gamma_{kpi} \le 0 \qquad \forall k \in K, p \in P_k, i \in N_p. \tag{56}$$

Similar to the SLM-A, we introduce the auxiliary variables, $\hat{q}_p = x_p \sum_{i\in N_p} q_i$ and $\hat{\gamma}_{kpi} = y_i \gamma_{kpi}$. We also introduce the parameter $q_{max}|N_p|$ representing the upper bound for $\hat{q}_p$, and introduce the parameter $-\lambda_k$ representing the lower bound for $\hat{\gamma}_{kpi}$. The linearization process is as follows:

$$0 \le \hat{q}_p \le x_p q_{max} |N_p| \qquad \forall k \in K, p \in P_k \tag{57}$$
$$(x_p - 1) q_{max} |N_p| + \sum_{i\in N_p} q_i \le \hat{q}_p \le \sum_{i\in N_p} q_i \qquad \forall k \in K, p \in P_k \tag{58}$$
$$-\lambda_k y_i \le \hat{\gamma}_{kpi} \le 0 \qquad \forall k \in K, p \in P_k, j \in N_k \tag{59}$$
$$\gamma_{kpi} \le \hat{\gamma}_{kpi} \le \gamma_{kpi} + (1 - y_i)\lambda_k \qquad \forall k \in K, p \in P_k, j \in N_k. \tag{60}$$

As in **Lemma 1**, the condition for the ϵ-optimality is as follows:

$$\left(\sum_{p\in P_k} t_p x_p + \sum_{p\in P_k} \hat{q}_p - \lambda_k z_k \right) - \left(\sum_{p\in P_k, i\in N_p} \hat{\gamma}_{kpi} + \theta_k \right) \le z_k \epsilon \qquad \forall k \in K \tag{61}$$
$$\sum_{p\in P_k} t_p x_p + \sum_{p\in P_k} \hat{q}_p - \lambda_k z_k \le 0 \qquad \forall k \in K. \tag{62}$$

We can reformulate the BLM-P into the following single-level mixed-integer linear programming model [SLM-P]:

[SLM-P] max	(9)	
s.t.	Constraints (3)–(8):	PWL queuing function
	Constraints (10), (11), (24):	Budget and flow capacity
	Constraints (25)–(30):	Follower's condition for the selected path
	Constraints (61)–(62):	Follower's condition for ϵ-optimality
	Constraints (52)–(56):	Follower's dual feasibility condition
	Constraints (57)–(60):	Linearization constraints.

6 Decomposition algorithm

Experiments on small transportation networks show that SLM-P outperforms SLM-A in terms of computational efficiency. However, neither model is tractable for large-scale networks. To address real-world large networks, we propose a logic-based decomposition algorithm (DA). The algorithm consists of two components that are solved iteratively, namely, a master problem and a subproblem. For clarity, the master problem (MP) generates candidate location vectors, while the subproblem (SP) evaluates each location vector under the original bilevel formulation. We define the MP as a refueling station location problem with routing (RSLP-R) by incorporating a budget constraint and additional time-related restrictions. This problem represents a relaxed version of the original problem that only considers the EV driving range and the time-threshold constraints. By solving the MP, we obtain candidate location vectors along with their corresponding optimal objective values in RSLP. For each location vector generated by the MP, we construct an SLM-P model as the SP to evaluate its optimal value under the bilevel formulation. Based on this evaluation, a logic-based cut is then added to the MP to guide the generation of new location vectors in subsequent iterations.

In principle, obtaining the global optimal solution of the bilevel problem would require generating and evaluating all possible location vectors. However, we can introduce a termination criterion that allows the algorithm to stop earlier. Specifically, the optimal value of the MP provides an overestimation of the bilevel optimal value and thus serves as an upper bound. Meanwhile, the best objective value obtained from all previously solved SPs is taken as a lower bound. The algorithm terminates when the upper bound from the MP coincides with the lower bound from the SP, indicating that the bilevel optimal solution has been identified.

6.1 Framework of the DA

In our decomposition scheme, the location decision vector $\overline{\mathbf{y}}$ obtained from the MP is passed to the SP, indicating which nodes are allowed to host charging facilities. Note that the MP does not determine the location decision of the bilevel problem. Instead, it only provides a set of candidate nodes $\Lambda(\overline{\mathbf{y}})$ along with their associated demands $\Phi(\overline{\mathbf{y}})$ and corresponding path sets $\Psi_k(\overline{\mathbf{y}})$. The SP must then determine the real location decision $\tilde{\mathbf{y}}$ from this candidate set, jointly with the deployment decisions $\mathbf{w}$, path selection $\mathbf{x}$, and coverage decisions $\mathbf{z}$ for the bilevel location problem. At each iteration, the optimal value of the SP is fed back to the MP via a logic-based cut.

The advantage of our approach is that, given the location vector $\overline{\mathbf{y}}$ from the MP, the nodes $i \in N$ with $\overline{y}_i = 0$ are excluded from SP. As a result, the SP only needs to consider a subset of nodes, demands, and their corresponding path sets with a reduced network. This substantially reduces the size and complexity of the bilevel problem. In addition, the MP generally provides a good upper bound for the bilevel problem. The flowchart of the DA is shown in Fig. 5.

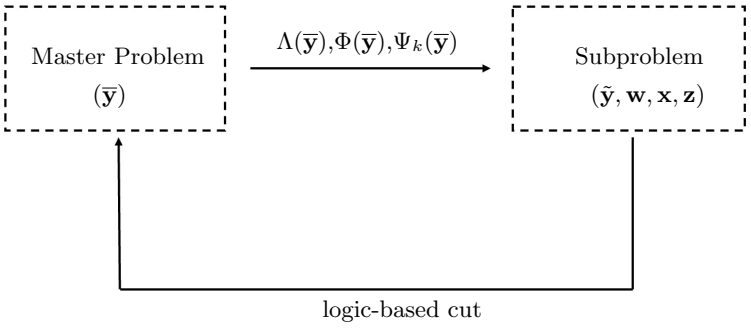


Figure 5: The flowchart of the DA

6.2 Master problem

The RSLP is defined as finding a set of nodes to open under a given budget to maximize the satisfied EV flow demand. For the RSLP, a demand is considered satisfied if there exists a path whose total travel time plus charging time is below the time threshold and that also satisfies the EV range constraint and Assumption (A2) along the way. The RSLP is a relaxation of the BCLP that ignores station capacity, EV path selection, and queuing time. However, it remains a computationally expensive problem for large networks. To the best of our knowledge, the branch-and-cut algorithm proposed by Arslan et al. (2019) with length-bounded cuts is the state-of-the-art method for solving large-scale RSLP-R. Therefore, we adapt their approach in our work.

For each demand $k \in K$, a subset of nodes $Q \subseteq N_k \setminus \{s_k, t_k\}$ is a length-bounded cut if, when these nodes are removed from the network $G_k = (N_k, A_k)$, all paths from s_k to t_k within the time threshold λ_k are disconnected. Let Γ_k be the set of all length-bounded cuts for each demand k. A cut Q is called a minimal length-bounded cut if none of the other cuts in Γ_k is a strict subset of Q.

The formulation of [MP] is as follows.

$$
\begin{aligned}
[\text{MP}] \quad \max \quad & \phi && && (63)\\
\text{s.t.} \quad & \sum_{i\in N}\sum_{j\in W} c_j w_{ij} \le B && && (64)\\
& \sum_{j\in W} w_{ij} = y_i && \forall i \in N && (65)\\
& z_k \le \sum_{i\in Q} y_i && \forall k \in K, Q \in \Gamma_k && (66)\\
& \phi \le \sum_{k\in K} f_k z_k && && (67)\\
& y_i, z_k, w_{ij} \in \{0,1\} && \forall k \in K, i \in N, j \in W && (68)\\
& + \quad \text{logic-based cuts from } \mathbf{SP}. && && (69)
\end{aligned}
$$

In constraints (66), the set of length-bounded cuts is exponential in size and is dynamically generated. For a given integer solution (y^*, z^*) of the [MP], we check whether each demand k with $z_k^* = 1$ can be covered with the selected location y^*. To this end, we construct the transformed network $G_k^*(N_k^*)$, where $N_k^* = \{s_k, t_k\} \cup \{i \in N_k : y_i^* = 1\}$ for each $k \in K$ with $z_k^* = 1$, and compute the shortest total time path from s_k to t_k in G_k^*. If this time exceeds the threshold λ_k, it indicates that the demand k cannot be covered, i.e., $z_k^* = 1$ does not hold, and we then easily obtain a length-bounded cut $Q_k^* = N_k / N_k^*$. Note that Q_k^* may not be a minimal length-bounded cut for demand k. To minimize its cardinality, we remove a node i from Q_k^* and add it to N_k^*. If, in the new network $G_k^*(N_k^* \cup \{i\})$,

there is still no shortest path from s_k to t_k that satisfies the time threshold λ_k, we repeat this process until we obtain a minimal length-bounded cut Q_k^{min}. We add the corresponding cut (66) to the [MP].

Although the problem setting considered by Kınay, Gzara, and Alumur (2023) differs from ours, both studies address bilevel location problems and employ a similar algorithmic framework. Both approaches iterate between a master problem that generates station-location decisions and a subproblem that evaluates the induced path-selection. Kınay, Gzara, and Alumur (2023) also observe that, for large-scale networks, even solving the location problem alone can be computationally expensive. To address this challenge, they propose a logic-based Benders algorithm (LBBA), in which the transformed network associated with each OD pair is partitioned into two connected components to generate feasibility cuts and deviation cuts, thereby accelerating the solution of the master problem.

Our algorithm differs from theirs primarily in the cut-generation strategy. Instead of relying on the partition-based feasibility and deviation cuts, we exploit more effective minimal length-bounded cuts tailored to the time-threshold structure of our problem. A detailed computational comparison between our algorithm and the LBBA of Kınay, Gzara, and Alumur (2023) on relevant benchmark instances is provided in Section 7.2.2.

6.2.1 Acceleration strategies

Since the MP is invoked repeatedly within the algorithmic framework, we introduce new acceleration strategies to further reduce its solution time. Note that minimal length-bounded cuts are dynamically generated during the separation phase. A natural idea is to explore whether effective cuts can be pre-generated and added to the model in advance, thereby accelerating the solution process. Based on this idea, we propose two types of minimal length-bounded cuts that can be pre-generated, which we refer to as **forward cuts** and **backward cuts**. For each OD pair k, let $\tilde{N}_k = N_k \setminus \{s_k, t_k\}$ denote the set of nodes in the transformed network G_k, excluding the origin and the destination nodes. We then sort the nodes in $\tilde{N}_k$ in non-decreasing order with respect to their distance from the origin node s_k (or equivalently, in non-increasing order with respect to their distance to the destination node t_k). The ordered sequence of nodes is given by $\Omega_k = \left\{n_1, n_2, \cdots, n_{|\tilde{N}_k|}\right\}$, such that $\delta_{s_k,n_1} \leq \delta_{s_k,n_2} \leq \cdots \leq \delta_{s_k,n_{|\tilde{N}_k|}}$, where Ω_k represents the index ordering of nodes in $\tilde{N}_k$ from the closest to the farthest relative to s_k.

Given the ordered sequence of nodes set Ω_k for OD pair k, we iteratively remove nodes from first to last to obtain a minimal forward length-bounded cut. Starting with $S = \Omega_k$, we perform the following steps: (i) remove the first node n_1 from S, and update $S = S \setminus \{n_1\}$; (ii) compute the shortest total time path between s_k and t_k in the expanded network $G_k(S)$, denoted as $\lambda_{G_k(S)}$; (iii) if $\lambda_{G_k(S)} \leq \lambda_k$, continue removing the next node. Repeat steps (i)-(iii) until $\lambda_{G_k(S)} > \lambda_k$ or all paths from s_k to t_k are disconnected. The resulting minimal forward cut set for OD pair k is denoted as $\overrightarrow{Q}_k^{min} = \tilde{N}_k \setminus S$. Similarly, we can obtain the minimal backward length-bounded cuts $\overleftarrow{Q}_k^{min}$ by sequentially removing nodes from right to left, following the same procedure. We incorporate the pre-generated cuts (66) for $\overleftarrow{Q}_k^{min}$ and $\overrightarrow{Q}_k^{min}$ for each OD pair k into the [MP] to speed up the solution time.

6.3 Subproblem

At each iteration, the MP can generate a location vector $\overline{\mathbf{y}}$ along with the corresponding coverage information $\overline{\mathbf{z}}$ of the OD pairs. Based on this information, the SP constructs an SLM-P model to determine the optimal objective function value of BCLP under the given location vector.

Recall that for each location vector $\overline{\mathbf{y}}$ from MP, $\Lambda(\overline{\mathbf{y}}) = \{i \in N \mid \overline{y}_i = 1\}$ is the set of available nodes with $\overline{y}_i = 1$; $\Phi(\overline{\mathbf{y}}) = \{k \in K \mid \overline{z}_k(\overline{\mathbf{y}}) = 1\}$ is the set of covered OD pairs in RSLP; $\Psi_k(\overline{\mathbf{y}}) = \{p \in P_k \mid t_p \leq \lambda_k \wedge N_p \subseteq \Lambda(\overline{\mathbf{y}})\}$ is the set of all available paths for OD pair $k \in \Phi(\overline{\mathbf{y}})$: a path is available if all the charging nodes along the path are available in $\overline{\mathbf{y}}$, and the sum of its travel time and charging time—excluding queuing time—remains within the threshold λ_k.

The [SP] can be formulated by using the [SLM-P] model and replacing N with $\Lambda(\overline{\mathbf{y}})$, K with $\Phi(\overline{\mathbf{y}})$, and P_k with $\Psi_k(\overline{\mathbf{y}})$, respectively. To avoid confusion with the location vector $\overline{\mathbf{y}}$ generated by the MP, we use the symbol $\tilde{\mathbf{y}}$ to represent the location variables in the SP.

$$
\begin{aligned}
[\text{SP}] \quad \max \quad & \sum_{k\in\Phi(\overline{\mathbf{y}})} f_k z_k && && (70)\\
\text{s.t.} \quad & \sum_{j\in W} w_{ij} = \tilde{y}_i && \forall i \in \Lambda(\overline{\mathbf{y}}) && (71)\\
& \sum_{i\in\Lambda(\overline{\mathbf{y}})}\sum_{j\in W} c_j w_{ij} \le B && && (72)\\
& \Delta_i = \sum_{k\in\Phi(\overline{\mathbf{y}})}\sum_{p\in\Psi_k(\overline{\mathbf{y}})} f_k x_p a_{pi} && \forall i \in \Lambda(\overline{\mathbf{y}}) && (73)\\
& \Delta_i \le \sum_{j\in W}\sum_{l\in I_j} \mu_j^l w_{ij} && \forall i \in \Lambda(\overline{\mathbf{y}}) && (74)\\
& \Delta_i = \sum_{j\in W}\sum_{l\in I_j} \mu_j^l b_{ij}^l && \forall i \in \Lambda(\overline{\mathbf{y}}) && (75)\\
& \sigma_{ij}^{|I_j|} \le \sigma_{ij}^{|I_j|-1} \le \ldots \le \sigma_{ij}^1 \le w_{ij} && \forall i \in \Lambda(\overline{\mathbf{y}}), j \in W && (76)\\
& \sigma_{ij}^{l+1} \le b_{ij}^l \le \sigma_{ij}^l && \forall i \in \Lambda(\overline{\mathbf{y}}), j \in W, l \in I_j && (77)\\
& q_i = \sum_{j\in W}\sum_{l\in I_j} \rho_j^l \mu_j^l b_{ij}^l && \forall i \in \Lambda(\overline{\mathbf{y}}) && (78)\\
& \sum_{p\in\Psi_k(\overline{\mathbf{y}})} x_p - z_k = 0 && \forall k \in \Phi(\overline{\mathbf{y}}) && (79)\\
& x_p \le \tilde{y}_i && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}), i \in N_p && (80)\\
& x_p \in \{0,1\} && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}) && (81)\\
& \left(\sum_{p\in\Psi_k(\overline{\mathbf{y}})} t_p x_p + \sum_{p\in\Psi_k(\overline{\mathbf{y}})} \hat{q}_p - \lambda_k z_k\right) \\
& \quad - \left(\sum_{p\in\Psi_k(\overline{\mathbf{y}}), i\in N_p} \hat{\gamma}_{kpi} + \theta_k\right) \le z_k \epsilon && \forall k \in \Phi(\overline{\mathbf{y}}) && (82)\\
& \sum_{p\in\Psi_k(\overline{\mathbf{y}})} t_p x_p + \sum_{p\in\Psi_k(\overline{\mathbf{y}})} \hat{q}_p - \lambda_k z_k \le 0 && \forall k \in \Phi(\overline{\mathbf{y}}) && (83)\\
& \pi_k + \sum_{i\in N_p} \gamma_{kpi} \le t_p + \sum_{i\in N_p} q_i && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}) && (84)\\
& \pi_k - \theta_k \ge \lambda_k && \forall k \in \Phi(\overline{\mathbf{y}}) && (85)\\
& \pi_k \text{ is free}, \theta_k \le 0, \gamma_{kpi} \le 0 && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}), i \in N_p && (86)\\
& 0 \le \hat{q}_p \le x_p q_{max} |N_p| && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}) && (87)\\
& (x_p - 1) q_{max} |N_p| + \sum_{i\in N_p} q_i \le \hat{q}_p \le \sum_{i\in N_p} q_i && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}) && (88)\\
& -\lambda_k \tilde{y}_i \le \hat{\gamma}_{kpi} \le 0 && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}), j \in N_k && (89)\\
& \gamma_{kpi} \le \hat{\gamma}_{kpi} \le \gamma_{kpi} + (1 - \tilde{y}_i)\lambda_k && \forall k \in \Phi(\overline{\mathbf{y}}), p \in \Psi_k(\overline{\mathbf{y}}), j \in N_k && (90)\\
& z_k \in \{0,1\}, \tilde{y}_i \in \{0,1\}, w_{ij} \in \{0,1\} && \forall i \in \Lambda(\overline{\mathbf{y}}), k \in \Phi(\overline{\mathbf{y}}), j \in W && (91)\\
& \sigma_{ij}^l \in \{0,1\}, 0 \le b_{ij}^l \le 1, \Delta_i \ge 0, q_i \ge 0 && \forall i \in \Lambda(\overline{\mathbf{y}}), j \in W, l \in I_j. && (92)
\end{aligned}
$$

Similar to SLM-P, constraints (71) to (74) represent the budget and flow capacity. Constraints (75) to (78) represent the PWL queuing function. Constraints (79) to (81) represent the follower's condition for the selected path. Constraints (82) to (83) represent the follower's condition for the ϵ-optimality. Constraints (84) to (86) represent the followers' dual feasibility condition. Constraints (87) to (90) represent the linearization.

Although SLM-P requires path enumeration, the [SP] is efficient in practice because the location vector $\bar{y}$ from [MP] induces a reduced network by excluding all nodes with $\bar{y}_i = 0$, thereby significantly reducing the set of candidate charging nodes and admissible paths. In the computational study in Section 7, we show that [SP] is difficult only in the hardest configurations, namely when the EV range is short, and the time-deviation is extreme (i.e., $\tau = 50\%$); in all other settings, it is solved efficiently.

After solving the [SP], we obtain the location decision $\tilde{\mathbf{y}}$, deployment decision $\mathbf{w}$, path selection decision $\mathbf{x}$, and the corresponding optimal objective function value of the BCLP. Let $\text{SP}[\overline{\mathbf{y}}]$ denote the optimal SP value for the location vector $\overline{\mathbf{y}}$. To guide the MP to produce a new location vector in subsequent iterations, we generate the following logic-based cut (93) and add it to the MP without affecting the optimal solution:

$$\phi \leq \left(\sum_{i:\overline{y}_i=1} (1 - y_i) + \sum_{i:\overline{y}_i=0} y_i \right) M + \text{SP}[\overline{\mathbf{y}}], \tag{93}$$

where the index sets $\{i : \overline{y}_i = 1\}$ and $\{i : \overline{y}_i = 0\}$ are defined with respect to the incumbent location vector $\overline{\mathbf{y}}$ of MP, and $M = \sum_{k \in K} f_k$. This cut enforces $\phi \leq \text{SP}[\overline{\mathbf{y}}]$ in MP if and only if location decision is $\overline{\mathbf{y}}$, thereby eliminating the previously evaluated vector from the search.

7 Numerical study

In this section, we report the results of extensive computational experiments to evaluate the performance of the proposed model and algorithm, and to investigate the effects of traffic flow, budget, time deviation level, and deployment configuration on the optimal solutions. All experiments are executed on the Narval HPC cluster (Digital Research Alliance of Canada), equipped with two AMD EPYC 7532 processors and 251 GB of RAM. Each instance is run on a single compute node for two hours. The algorithms are implemented in Java and use the Gurobi Optimizer (v12.0.0).

7.1 Datasets and experimental settings

We consider three sets of test instances: a 25-node small-scale network, a large-scale road network in California, and a real-world road network in Québec. The 25-node small-scale network, denoted as *N25*, was introduced in Simchi-Levi and Berman (1988) and serves as a small benchmark. In the original *N25* network, the length of each arc was labeled. We multiply the length of each arc by 10 km (Zhang et al. 2016). For example, the distance between node 1 and node 2 is 4 in *N25*, which is considered as 40 km in our experiments. The California road network, denoted as *CA339*, originates from Yıldız, Arslan, and Karaşan (2016) and contains 339 nodes; it is commonly used as a large-scale benchmark to evaluate algorithmic performance. The third instance, denoted as *QC405*, represents a real-world case based on data from Hydro-Québec, comprising 405 nodes in the Québec road network. The detailed characteristics of these instances, such as the number of potential O/D nodes (i.e., generating OD pairs), the number of arcs, and the average and maximum arc lengths, are presented in Table 1. The road networks are shown in Fig. B1 and Fig.B2 in Appendix B.

Table 1: The characteristics of networks

Network	# Nodes	# O/D nodes	# Arcs	# OD pairs	Avg. arc length (km)	Max. arc length (km)
N25	25	25	86	300	46.00	90.00
CA339	339	57	1234	1167	18.96	95.21
QC405	405	88	1003	2669	10.02	48.07

Regarding the EV path selection, ϵ is set to 5 minutes, meaning that EV drivers choose paths whose total disutility is at most 5 minutes worse than the best available disutility. The EVs are assumed to

travel at 80 km/h, and Level-3 fast chargers with an exponential service time of 30 minutes are used in all deployment configurations (Xie and Lin 2021). All time quantities are measured in minutes. Following Kınay, Gzara, and Alumur (2023), we set the construction cost of each charger to \$75,000, and the construction cost of each station to \$150,000.

To approximate the $M/M/c$ queuing system, we construct a three-segment piecewise-linear function for each deployment configuration as shown in Fig. 1. We truncate the queuing time curve to the range of 0 to 30 minutes. We take the queuing time from 0 to 1 minute in the $M/M/c$ function as the first segment of the PWL function, in which the slope is set to 0 and the queuing time is fixed to 0; the queuing time from 1 to 10 minutes as the second segment; and the queuing time from 10 to 30 minutes as the third segment. The details are shown in Figures B3 to B5 in Appendix B.

Note that our model and solution approach are capable of handling multiple EV demand classes associated with the same OD pair, characterized by different flow volumes, driving ranges, and time deviation tolerances. However, as in previous studies, we assume that all demand classes have the same EV driving range and the same time deviation tolerance, which we denote by $R = R_k$ and $\tau_T = \tau_k$ for all $k \in K$.

According to the Assumption (A2), we assume that OD pairs with distances shorter than half of the EV range do not require intermediate charging, and only OD pairs with distances exceeding half of the EV range are considered. Therefore, different EV range settings lead to different instances. As real OD-pair flow data are unavailable, we assume an average traffic of H EVs per hour across the entire network. The flow H_{ij} between any two OD nodes (i, j) is generated using a gravity model based on distance δ_{ij} and population size associated with each node (ω_i and ω_j): $H_{ij} = \omega_i \omega_j / \delta_{ij}^{1.5}$, and the hourly EV flow rate is calculated by $H_{ij}^{\mathrm{EV}} = H \frac{H_{ij}}{\sum_{i \neq j} H_{ij}}$ (Zhang et al. 2016). For each OD pair k (from node i to node j), we define its Poisson OD flow rate as $f_k = H_{ij}^{\mathrm{EV}}$. To facilitate computation and presentation of the results, we normalize the flows of OD pairs and express the objective function in percentage terms.

7.2 Computational performance and benchmark comparisons

We first demonstrate that the proposed acceleration strategies yield substantial improvements in the computational efficiency of solving the MP. We then show that our DA framework is a general solution approach that can be applied to solve a closely related bilevel location problem and compare it with a state-of-the-art algorithm.

7.2.1 Computational performance of the acceleration strategies

To demonstrate the effectiveness of the proposed acceleration strategies in improving the computational performance of the DA algorithm, we conduct experiments with and without these strategies enabled. Figures F1 and F2 in Appendix F summarize the impact of the proposed acceleration strategies on the computational performance of the DA algorithm. For both the *CA339* and *QC405* networks, the accelerated variant consistently reduces the average MP runtime per iteration across all EV ranges and time-deviation settings.

In the *CA339* network, the runtime reductions are substantial: for example, under $R = 200$ km and $\tau_T = 50\%$, the average MP solution time decreases from 22.54 seconds to 6.4 seconds. Similar improvements are observed for $R = 250$ km and $R = 300$ km. In the *QC405* network, the improvements remain significant. Under $R = 200$ km and $\tau_T = 50\%$, the runtime drops from 35.66 seconds to 6.29 seconds, while for larger EV ranges ($R = 300$ km and $R = 400$ km) the reductions exceed 70% in most cases.

7.2.2 Computational comparison with Kınay, Gzara, and Alumur (2023)

Our DA framework is a general solution approach for bilevel location problems. Its MP is responsible for generating high-quality location decisions, while its SP evaluates the generated location decisions. To demonstrate the generality of the DA framework, we apply it to solve the related bilevel location problem of Kınay, Gzara, and Alumur (2023). In their setting, each station is required to satisfy a prescribed queuing service level, and the objective is to cover all OD pairs while minimizing the total investment cost in stations and chargers. Although the detailed problem setting of Kınay, Gzara, and Alumur (2023) differs from ours in several aspects—including the objective function, the queuing function, and the EV path selection logic—our DA algorithm can be adapted accordingly to solve their problem. The details of these modifications for DA are provided in Appendix G.

Kınay, Gzara, and Alumur (2023) generated instances by considering different queuing service levels, values of R, and distance-deviation. Two networks are studied, the *CA339* network and the *USEast420* network, resulting in a total of 64 instances. They proposed a logic-based Benders algorithm (LBBA) to solve these instances within one hour. We apply the adapted DA algorithm to solve the same set of 64 instances under the same one-hour time limit.

We obtained and ran the source code of Kınay, Gzara, and Alumur (2023) in our computational environment, and observed that the results were generally worse than those reported in Kınay, Gzara, and Alumur (2023). This discrepancy is mainly due to the fact that our CPU is less powerful than the one used in their experiments. Therefore, when reporting the performance of LBBA, we directly use the results reported in Kınay, Gzara, and Alumur (2023). The detailed comparison results between LBBA and DA are reported in Tables G1 and G2 in Appendix G. Here, "Gap" denotes the relative difference between the results obtained by the two algorithms. A positive value indicates an improvement (i.e., a lower total cost), whereas a negative value indicates a deterioration. As shown in the tables, our algorithm outperforms LBBA on the vast majority of instances, even though our experiments are conducted on a less powerful computational platform. The maximum cost reduction reaches 7.48%. Among the 64 instances, our algorithm obtains a better solution on 49 instances, the same solution on 8 instances, and a worse solution on 7 instances. The main advantage of the DA algorithm over LBBA is its ability to solve the location problem more efficiently, enabling high-frequency iterative exploration. For example, on the *USEast420* network with $R = 400$ km and a 50% distance-deviation, LBBA performs only one iteration within the one-hour time limit, whereas our DA algorithm completes more than 4,000 iterations within the same time limit. As noted in Kınay, Gzara, and Alumur (2023), LBBA struggles to solve the location problem when the deviation tolerance is high (e.g., 50%), resulting in very slow progress. In contrast, our DA algorithm remains computationally efficient even under such extreme settings, frequently executing thousands of iterations within one hour. These results highlight the computational efficiency of the proposed DA algorithm.

7.3 N25 network

In the *N25* network, each node represents a potential charging station location. Based on realistic settings, we define three deployment configurations that correspond to installing 4, 6, or 8 chargers, respectively. We assume that the EV traffic passing through the entire *N25* network is $H = 50$ EVs/hour. Experiments are conducted under time deviation levels of 0%, 10%, 25%, and 50%, with EV ranges of 150 km, 200 km, and 250 km, and budget levels of 0.9, 1.8, 2.7, 3.6, and 4.5 million dollars. Since only OD pairs with distances exceeding half of the EV range are considered, the number of OD pairs is 236, 211, and 167 for $R = 150$, 200, and 250 km, respectively. In total, 60 test instances are generated.

We conduct experiments on each instance using the model SLM-A, model SLM-P, and the DA algorithm. The maximum runtime for solving the SLM-P and SLM-A models in Gurobi is set to one hour. The detailed experimental results are presented in Tables C1 to C3 in the Appendix C, which show that all three methods are capable of finding optimal solutions.

The comparison of the computational performance of the SLM-A, the SLM-P, and the DA are shown in Fig. 6. The DA algorithm demonstrates a remarkable computational advantage over both mathematical models. Except for the case with $\tau_T = 0\%$, the average runtime of the DA is significantly lower than that of both models under all other settings. Moreover, as τ_T increases, the computational advantage of DA becomes even more pronounced. In the case with $R = 150$ km, the average runtime of DA is only 10.0 s, which is approximately 25 times faster than SLM-A (253.08 s) and 6 times faster than SLM-P (60.36 s). The same trend is observed for $R = 200$ km and $R = 250$ km. These results indicate that DA has clear advantages in computational speed and scalability. In addition, between the two mathematical models, SLM-P performs significantly better than SLM-A. On average, the runtime of SLM-P is about one-fifth of that of SLM-A.

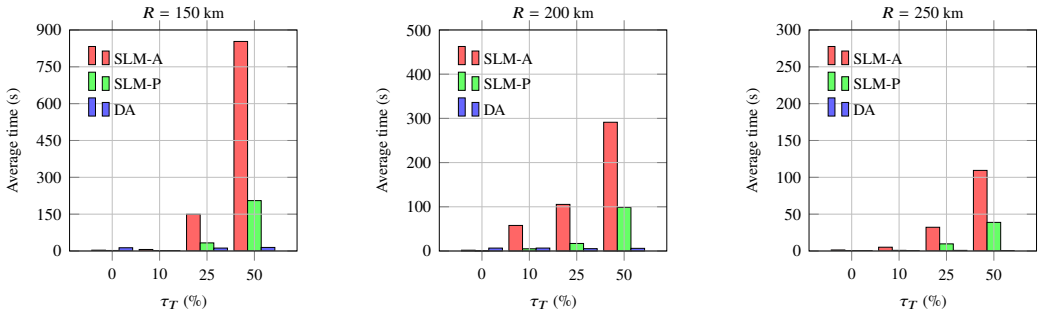


Figure 6: The average runtime of the proposed model and algorithm on the N25

Figures 7–8 suggest that higher budgets are associated with increased covered OD flow and reduced average queuing times. When the time deviation is set to a high level, such as 20% or 50%, low-budget scenarios with only a few opened stations tend to produce long expected queuing times. Under a fixed budget, a higher time deviation tolerance significantly increases the covered OD flow but also results in longer queuing times, which is an expected outcome. We observe that under some settings with low budgets and large deviation levels, the average expected queuing time among the open charging stations can reach up to 15 minutes. In addition, a larger EV range greatly reduces the required investment to achieve full OD pair coverage. For instance, when $R = 150$ km, even with a 4.5 million budget and a 50% time deviation, only 99.95% of the OD flow can be covered. However, when $R = 200$ km, full coverage can be achieved with just a 3.6 million budget and a 25% time deviation; and for $R = 250$ km, only 2.7 million is sufficient to cover all OD pairs fully.

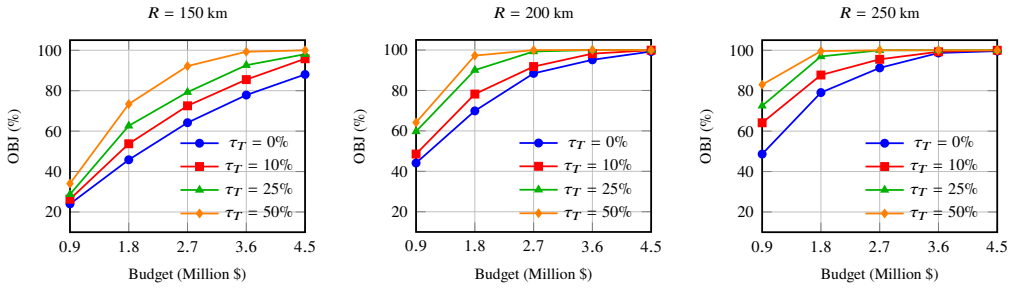


Figure 7: Covered OD flow of the N25 instance under different budgets and time deviation settings

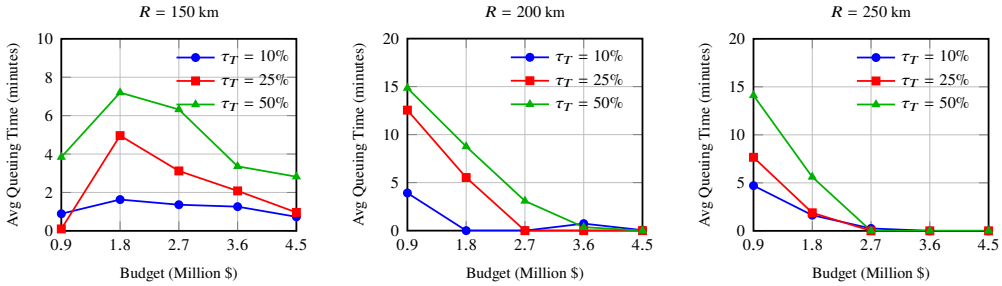


Figure 8: Average queuing time among open nodes of the N25 instance under different settings

To examine the impact of ϵ on the objective value, Table C4 in Appendix C compares the results obtained under $\epsilon = 0$ min, where EV drivers are restricted to select strictly optimal paths, with those under $\epsilon = 5$ min. It can be observed that for the vast majority of instances, the objective values remain identical under the two settings. However, in a few cases, the covered flow decreases when ϵ is set to zero, as highlighted in bold. For example, consider the instance with $R = 150$ km, $\tau_T = 25\%$, and $B = 3.6$ million. The optimal location decision obtained under $\epsilon = 5$ min does not admit a feasible EV driver response when $\epsilon = 0$ min, making it impossible to achieve an objective value of 92.61%. As a result, an alternative location decision must be selected, yielding a reduced objective value of at most 90.71%. This result also confirms our analysis in Section 3.3, showing that ϵ-optimality guarantees the existence of a stable follower response for the leader decisions.

7.4 CA339 California network

The *CA339* network consists of 339 nodes, which represent urban population centers, and 57 centers with populations greater than 50,000 are considered as OD nodes. Therefore, unlike the *N25* network, the nodes in the *CA339* network no longer represent charging stations but rather correspond to regions or cities. EV Infrastructure News reports that California currently has over 2.45 million registered electric vehicles.[5] Accordingly, we set the total EV traffic volume to $H = 10{,}000$ EVs/h and use this value to distribute the EV flow per hour for each OD pair based on the gravity model. Based on the data from California Energy Commission, California has in excess of 178,000 public EV chargers.[6] Accordingly, we consider three deployment configurations, in which each node (population center or city) can be installed with either 150, 200, or 250 chargers. We consider time deviation levels of 0%, 10%, 25%, and 50%, with EV ranges of 200, 250, and 300 km, and budget levels of 24, 36, 48, 60, and 72 million dollars. The number of OD pairs is 806, 670, and 542 for $R = 200$, 250, and 300 km, respectively.

Both models, SLM-A and SLM-P, fail to solve the large-scale instance *CA339*. Therefore, we conduct the experiments using the DA algorithm, setting a maximum runtime limit of two hours. Table 2 shows the summary computational results of the DA algorithm on the *CA339* instance. The Opt Gap in the table represents the gap between the MP and SP in the DA algorithm. The results indicate that the DA algorithm obtains high-quality solutions with very small optimality gaps, solving 52 out of 60 instances optimally. The average optimality gap is 0.80% for the cases with $R = 200$ km, 0.00% for $R = 250$ km, and 0.12% for $R = 300$ km. The average optimality gap across all instances is 0.31%. The detailed experimental results are provided in Tables D1 to D3 in Appendix D. As shown in our experiments, the average queuing times remain very small across all scenarios, enabling the DA algorithm to find the optimal solution for most instances. The reason for the short queuing times is the nature of the $M/M/c$ queuing function used to model each charging node. Given the deployment

[5] https://www.evinfrastructurenews.com/ev-networks/california-sets-ev-sales-record-in-q3-2025-ahead-of-tax-credit-expiry

[6] https://www.energy.ca.gov/news/2025-03/california-exceeds-178000-electric-vehicle-chargers

configurations of 150, 200, or 250 chargers, the $M/M/c$ curves in Fig. B4 show that the queuing time remains zero until the flow rate reaches approximately 95% of the service capacity.

Table 2: Computational results of the DA algorithm on the CA339 instance

τ_T	R=200 km			R=250 km			R=300 km		
	#Opt	Time (s)	Opt Gap	#Opt	Time (s)	Opt Gap	#Opt	Time (s)	Opt Gap
0%	5/5	0.66	0.00%	5/5	0.36	0.00%	5/5	0.34	0.00%
10%	5/5	46.98	0.00%	5/5	1.43	0.00%	5/5	33.03	0.00%
25%	2/5	5530.26	0.44%	5/5	18.75	0.00%	5/5	1428.21	0.00%
50%	1/5	5768.03	2.76%	5/5	361.21	0.00%	4/5	1445.10	0.49%
Tol	13/20	2836.48	0.80%	20/20	95.44	0.00%	19/20	726.67	0.12%

Fig. 9 shows the covered OD flow of the *CA339* network under different budgets and time deviation settings. Due to the long and narrow geographical structure of the *CA339* network and the strong concentration of OD nodes in the central region, the experimental results indicate that only a small number of city nodes need to host charging facilities to cover all OD pairs. For example, in the instance with an EV range of 200 km, deploying 900 chargers at only six nodes is sufficient to achieve full coverage. When the EV range increases to 250 km or 300 km, the required infrastructure becomes even smaller, with only three to four nodes hosting 500 to 600 chargers. In addition, the covered OD flow increases rapidly as the time deviation grows. Based on the numerical results, increasing the EV driving range has a significantly stronger effect on improving coverage than increasing the deployment budget.

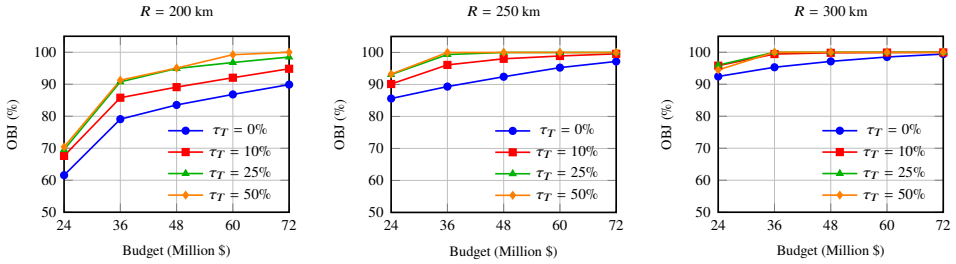


Figure 9: Covered OD flow of the CA339 network under different budgets and time deviation settings

We also report the average queuing time under different budget and time deviation settings in Fig. 10. Queuing appears in the case with a relatively short EV range of $R = 200$ km. When the time deviation is 10%, the average queuing time is about one minute. For a deviation of 25%, the average queuing time increases to around two minutes. When the deviation reaches 50%, the average queuing time is approximately 7 minutes. In contrast, for larger EV ranges of $R = 250$ km and $R = 300$ km, both the average and maximum queuing times are very close to zero across all budgets and deviation levels, indicating that congestion at charging nodes becomes negligible once the driving range is sufficiently long.

Fig. 11 reports the average runtime of the MP and SP per iteration under different time deviation settings. Overall, both MP and SP exhibit an increasing trend in runtime as the time deviation grows, indicating that larger deviation allowances generally make the MP and SP more complex to solve. Nevertheless, the overall computation time remains very short — even under the most extreme setting of $\tau_T = 50\%$, both MP and SP can be solved within a few seconds on average. This demonstrates the high computational efficiency of the proposed DA algorithm, which can handle multiple iterations effectively even for relatively challenging configurations.

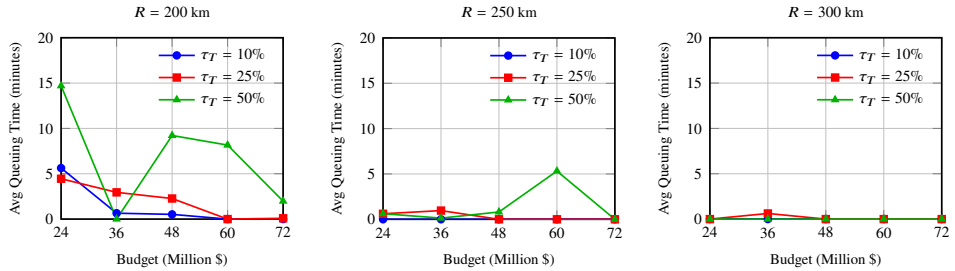


Figure 10: Average queuing time among open nodes of the CA339 network under different settings

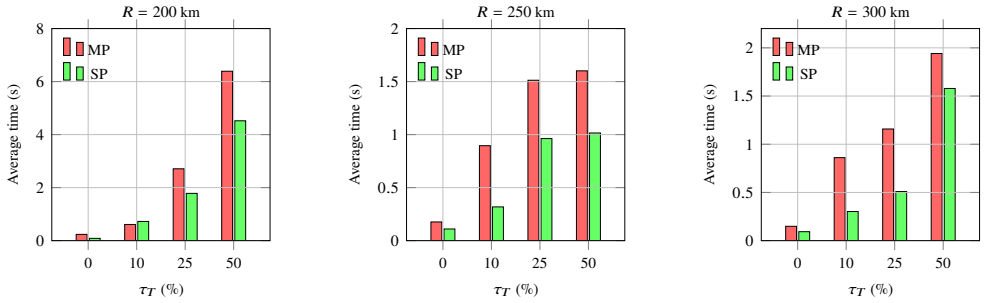


Figure 11: Average runtime of the MP and SP per iteration in the DA algorithm of the CA339 instance

We finally analyze all deployment solutions obtained on the *CA339* network and identify the six nodes that appear the most frequently in all solutions. These high-frequency nodes can be interpreted as hotspot charging nodes, as shown in Fig. F3 in Appendix F.

7.5 Case study: QC405 Québec network

Our industrial collaborator, Hydro-Québec, provided a realistic and representative dataset of the Québec network. This network is primarily composed of road-junction nodes and population-center nodes, the latter being treated as O/D nodes. The populations of these centers range from 6,000 to 2 million. The original Québec network contains 94 O/D nodes, 5,673 nodes, 9,398 arcs, and 5,992 OD pairs. The dataset also provides the total traffic volume for each OD pair during the summer period from 12:00 to 18:00. For experimental purposes, nodes located within 10 km of each other are merged. After simplification, the resulting network, referred to as *QC405*, contains 88 O/D nodes, 405 nodes, 1,003 arcs, and 2669 OD pairs. Similar to the settings used for the *N25* and *CA339* networks, we remove all OD pairs whose shortest-path distance is less than $R/2$. Because drivers typically do not travel more than 500 km in a day without becoming fatigued, we set an upper limit of 500 km and remove all OD pairs whose shortest-path distance exceeds this threshold.

As a result, the nodes in the *QC405* network should be interpreted as road-network access points and are suitable for deploying medium-scale charging facilities. To ensure realistic modeling, we consider deployment configurations of 15, 30, and 45 chargers per site. We consider time deviation levels of 0%, 10%, 25%, and 50%, and budget levels of 20, 30, 40, 50, and 60 million dollars. Given that EV ranges in summer typically fall between 200 km and 400 km, we consider $R = 200$, 300, and 400 km in our experiments, and the number of OD pairs is 1326, 1322, and 1182, respectively.

We conduct experiments on the 60 instances using the DA algorithm, with a maximum runtime limit of two hours. The summary results are presented in Table 3. It shows that the *QC405* network is more difficult to solve than the *CA339* network because it contains more O/D nodes and OD pairs. Even so, the DA algorithm can obtain optimal solutions for 39 out of the 60 test instances. For

$R = 200$ km, the average optimality gap is 4.25%; for $R = 300$ km, it is 0.70%; and for $R = 400$ km, it is 0.08%. The average optimality gap across all instances is 1.67%. The detailed results are presented in Appendix E.

Table 3: Computational results of the DA algorithm on the QC405 instance

τ_T	R=200 km			R=300 km			R=400 km		
	#Opt	Time	Opt Gap	#Opt	Time	Opt Gap	#Opt	Time	Opt Gap
0%	3/5	2963.47	1.00%	5/5	1.88	0.00%	5/5	47.53	0.00%
10%	1/5	5886.56	3.45%	4/5	1498.96	0.12%	3/5	2888.27	0.16%
25%	1/5	6686.56	6.88%	3/5	3041.15	1.56%	4/5	1725.48	0.13%
50%	1/5	6203.59	5.68%	3/5	3170.62	1.12%	4/5	1804.52	0.02%
Avg	8/20	5435.04	4.25%	15/20	1928.15	0.70%	16/20	1616.45	0.08%

We observe that, due to the wide geographical distribution of population centers in the Québec network, covering all OD pairs requires deploying charging facilities at many nodes. Fig. F5 corresponds to an additional illustrative scenario outside the main experimental setup. It demonstrates that covering all OD flows requires deploying 1,785 chargers at 102 nodes. Fig. F6 demonstrates the deployment solution under a full-coverage setting when $R = 200$ km, and a time deviation of 25%. A total of 465 chargers are deployed across 29 nodes. Fig. F7 demonstrates the deployment solution under a full-coverage setting when $R = 400$ km, and a time deviation of 10%. A total of 585 chargers are deployed across 37 nodes. Fig. F4 shows the top ten charging nodes with the highest selection frequency across all deployment solutions, referred to as hotspot charging locations.

Fig. 12 illustrates how the objective value changes with increasing time deviation and budget levels. Because a large number of nodes are concentrated in the southwest part of the network, increasing the EV range and allowing time deviation can significantly improve the OD pair coverage. For example, when the EV range is 300 km and the time deviation is $\tau_T = 10\%$, deploying 465 chargers at only 30 nodes is sufficient to cover nearly all OD pairs.

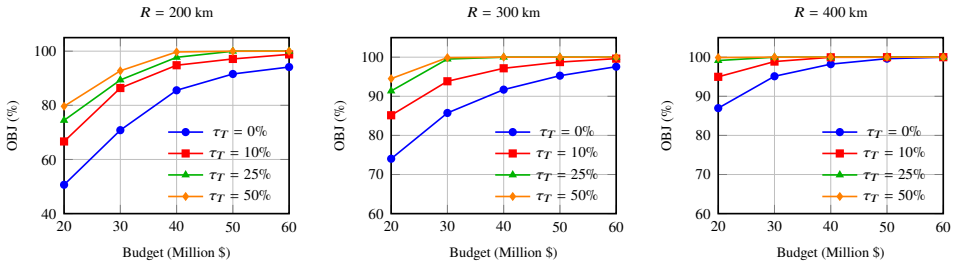


Figure 12: The objective value of QC405 instance under different settings

Figs. 13 and 14 show that increasing the budget consistently reduces the average queuing time and the maximum queuing time among open nodes. This improvement is most pronounced for high time deviation cases (e.g., $\tau_T = 50\%$), where the queuing time decreases rapidly when the budget increases, especially for smaller EV ranges such as $R = 200$ km. When the budget reaches 50 million dollars, the average queuing time becomes negligible for all settings.

Fig. 15 illustrates the reason behind the computational difficulty of the *QC405* network. As shown, the MP in the DA algorithm still can be solved very quickly, within seconds at each iteration, whereas the SP constitutes the most time-consuming component. For example, under $R = 200$ km with time-deviation settings of 25% and 50%, the average SP solving time ranges from 300 to 600 seconds per iteration. This is because the location vectors produced by the MP typically open 30 to 40 nodes, which drastically increases the number of paths that must be enumerated when constructing the SP,

thereby increasing the computational burden. The situation improves for larger EV ranges. Under $R = 300$ km, the average SP solving time drops to approximately 125 to 250 seconds for time-deviation levels of 25% and 50%. When $R = 400$ km, the average SP solving time further decreases to below one minute.

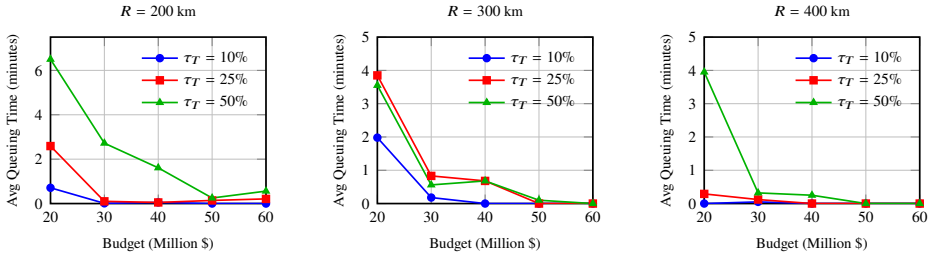


Figure 13: Average queuing time among open nodes of QC405 network under different settings

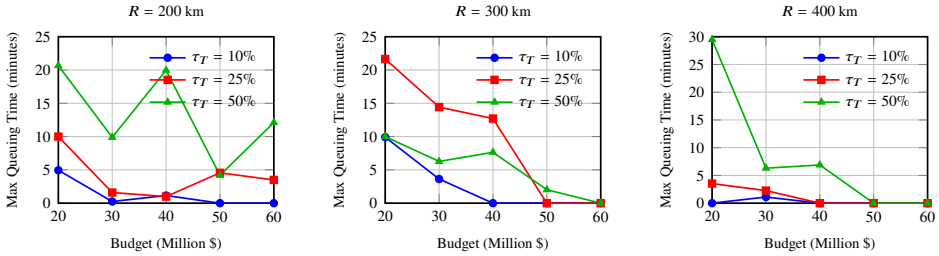


Figure 14: Maximum queuing time among open nodes of QC405 network under different settings

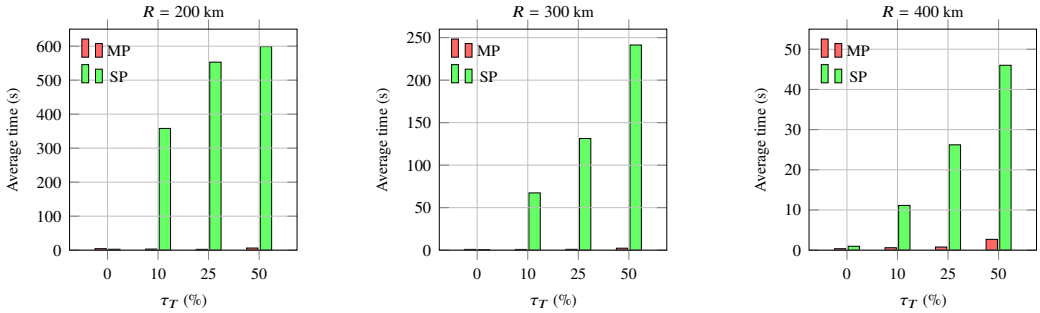


Figure 15: Average runtime of MP and SP per iteration in the DA algorithm of QC405 instance

To demonstrate the value of explicitly incorporating queuing time in EV path selection, we conduct the following experiment on *QC405* network.

Experiment 1 (NoQ–NoCap). We first solve a simplified variant of the original problem by removing the queuing-time term from the disutility function (1) and dropping the charging-capacity constraints. We refer to this setting as NoQ–NoCap. The solution provides a leader decision (locations and deployment) and the corresponding EV path selections, where drivers ignore queuing. We then fix the NoQ–NoCap leader decision and re-evaluate the instance under the original setting, which accounts for both queuing and capacity. Experiments use $R = 200$ km, $\tau \in \{0\%, 10\%, 25\%, 50\%\}$, and $B \in \{20, 30, 40, 50, 60\}$ M\$. Table 4 reports the NoQ–NoCap solutions and their re-evaluation. Here, $\#N$ is the number of opened nodes; #CapVio denotes the number of opened nodes obtained under the NoQ-NoCap setting whose induced flow exceeds the charging capacity when capacity constraints

are considered (unstable $M/M/c$, hence infinite queuing time); #OD_N denotes the number of covered OD pairs under NoQ-NoCap setting that become uncovered under the original setting (i.e., when queuing time is considered); Max q denotes the maximum finite queuing time among the opened nodes without capacity violation (in minutes) for the NoQ-NoCap solution, computed using the $M/M/c$ queuing function (2). Obj_Fix is the covered flow obtained by fixing the location decisions from the NoQ-NoCap setting and re-solving the SP under the original problem setting. Obj_Org is the covered flow obtained by the DA algorithm under the original setting, and Gap1 denotes the difference between them.

Table 4 shows that ignoring queuing produces leader decisions that are suboptimal under the original setting. As indicated by Gap1, the objective gap can be as large as 53.11%. Moreover, even without capacity violations, induced queuing times can be large enough to break time-deviation feasibility. For instance, under $\tau_T = 50\%$ and $B = 30$ M$, the NoQ–NoCap solution yields a maximum queuing time of 76.35 minutes, 9 out of the 23 opened nodes exceed their charging capacity, and the resulting queuing time render 1,259 OD pairs— covered under the NoQ-NoCap setting—no longer coverable under originial setting.

Table 4: Evaluation of the NoQ–NoCap Solutions and the NoQ Solutions

τ_T (%)	B (Mil $)	NoQ-NoCap Setting					NoQ Setting					Org. Setting	Gap1 (%)	Gap2 (%)
		# N	# CapVio	# OD_N	Max q (minutes)	Obj_Fix (%)	# N	# OD_N	# ϵ-Vio	Max q (minutes)	Obj_Fix (%)	Obj_Org (%)		
0	20	15	1	360	1.29	42.09	14	296	0	16.85	50.65	50.65	16.90	0.00
	30	23	1	541	5.72	67.51	23	643	0	26.76	70.82	70.82	4.67	0.00
	40	31	2	579	6.64	81.44	31	558	0	28.86	84.68	85.55	4.80	1.02
	50	39	1	495	9.06	91.57	39	359	0	20.94	91.57	91.57	0.00	0.00
	60	47	1	207	0.59	94.14	47	163	0	4.06	94.14	94.14	0.00	0.00
10	20	15	4	806	73.9	66.49	13	356	17	29.42	66.37	66.65	0.23	0.42
	30	23	4	710	35.31	75.65	23	648	106	28.69	71.17	86.39	12.43	17.61
	40	31	2	452	37.92	93.95	31	346	93	26.09	87.56	94.80	0.89	7.63
	50	39	1	396	14.76	76.98	39	405	137	28.77	73.26	97.13	20.74	24.58
	60	47	1	305	4.74	98.81	47	76	166	25.33	98.81	98.81	0.00	0.00
25	20	15	7	1139	1.06	71.15	11	43	44	17.28	71.25	74.45	4.43	4.30
	30	23	6	1090	38.10	52.87	20	141	266	28.16	85.83	89.36	40.83	3.95
	40	31	7	1098	2.88	61.40	26	36	205	25.67	97.37	97.73	37.18	0.37
	50	39	7	1140	34.31	81.58	34	46	155	28.97	82.19	99.98	18.40	17.79
	60	47	4	855	2.71	80.67	41	4	32	4.22	99.99	100.00	19.33	0.01
50	20	15	7	1233	33.87	69.67	9	4	15	11.91	73.39	79.65	12.53	7.86
	30	23	9	1259	76.35	58.10	16	86	309	29.73	68.80	92.75	37.36	25.82
	40	31	7	1168	57.74	65.12	25	14	102	27.79	99.38	99.71	34.69	0.33
	50	39	7	1224	2.91	81.51	31	1	80	10.55	99.98	99.98	18.47	0.00
	60	47	5	1132	11.04	46.89	37	7	146	26.13	88.69	100.00	53.11	11.31

Max q: maximum queuing time computed using the $M/M/c$ queuing function (2).

Experiment 2 (NoQ). We next solve a setting where queuing is excluded from the disutility function but capacity is enforced (NoQ). We solve the problem under the NoQ setting, and fix the leader decision and re-evaluate it under the original setting (Table 4). Gap2 represents the gap between Obj_Fix under the NoQ setting and Obj_Org. Enforcing capacity substantially reduces infeasibility and makes the objective gap small in most instances. However, the path selections obtained under NoQ can still be inconsistent with the queuing-aware response. Column #ϵ-Vio reports the number of selected paths that violate ϵ-optimality once queuing is considered. We observe many violations, and in some instances NoQ and the original setting achieve the same objective value while inducing different covered OD sets and path selections. For example, under $\tau = 10\%$ and $B = 60$ M$, both settings attain 98.81%, yet 76 OD pairs covered under NoQ become uncovered under the original setting, and 166 selected paths violate ϵ-optimality. We also observe that, under the NoQ setting, the

maximum queuing time (Max q) does not exceed 29.73 minutes. This result justifies truncating the $M/M/c$ queuing time curve to the 0–30 minute range when constructing the PWL approximation, as this range is sufficient to capture the vast majority of practical cases with respect to charging capacity.

Overall, the results of the NoQ-NoCap and NoQ settings confirm the value of incorporating queuing effects into EV path selection. When queuing time is ignored in the path choice process, the resulting location decisions and objective values may be suboptimal under realistic operating conditions. Moreover, these two settings can lead to a substantial amount of uncovered OD flow and paths whose total travel time exceeds the prescribed time threshold.

8 Conclusion

This paper introduced a new bilevel charging facility location problem in which EV drivers choose routes based on a disutility function that incorporates travel time, charging time, and congestion-induced queuing delays modeled through an $M/M/c$ system. At the leader level, a planner determines the locations and deployment configurations of fast-charging stations under a given budget, while at the follower level, EV drivers respond to these decisions through path choices. This creates an interplay between path selection decisions and congestion at charging nodes, leading to the nonexistence of an optimal solution, which we handle via ϵ-optimality.

We developed two bilevel mathematical formulations—an arc-based and a path-based model—and derived equivalent single-level MILP reformulations using strong duality and piecewise-linear approximations of the queuing function. Although these reformulations allow exact solution by general-purpose solvers, they do not scale to realistically sized networks.

To address this challenge, we proposed an exact decomposition algorithm (DA) that alternates between a RSLP master problem and a path-based evaluation subproblem. The DA is further strengthened by two families of pre-generated minimal length-bounded cuts and several acceleration strategies that make the state-of-the-art RSLP-R solver on average, ten times faster, enabling the master problem to be solved within seconds.

Extensive computational experiments on the *N25*, *CA339*, and the case study of *QC405* networks demonstrate the scalability and effectiveness of the proposed approach. The DA achieves more than an order-of-magnitude speedup over the single-level MILP models, solves 91 out of 120 CA339 and QC405 network instances to optimality within two hours, and achieves average optimality gaps of 0.31% on *CA339* and 1.67% on *QC405*. The bilevel optimal solutions reveal clear interactions among budget, EV range, network geometry, and congestion patterns. Longer EV ranges and moderate time deviation threshold significantly increase OD coverage while reducing the number of required charging facilities; however, insufficient charger deployment may create localized congestion that sharply increases expected queuing delays. Finally, we applied the DA algorithm to a benchmark from another related problem and compared it against a state-of-the-art method. The results demonstrate the superior performance of our approach, with the DA algorithm finding a large number of improved solutions.

In general, this work provides a scalable and exact algorithmic framework for optimizing the location of charging stations under realistic congestion and routing behavior, offering actionable insights for the planning of fast-charging infrastructure in large transportation networks.

Appendix A Summary of symbols

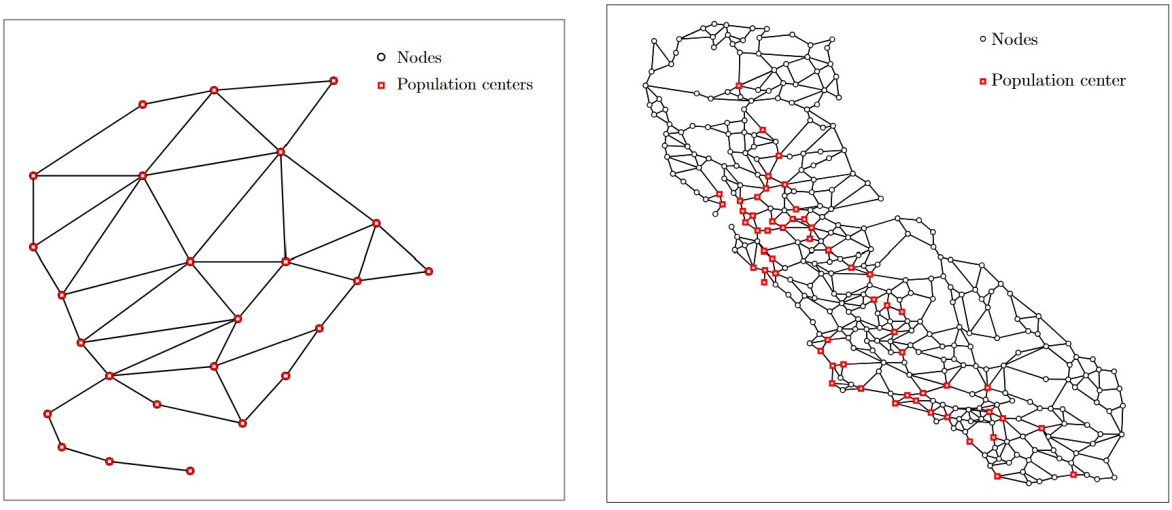
Table A1: General sets, parameters, and decision variables

Set	Description
K	set of OD pairs
A	set of arcs
N	set of nodes
A_k	set of arcs in the transformed network for OD pair k
N_k	set of nodes in the transformed network for OD pair k
P_k	set of all possible paths for OD pair k
N_p	set of charging nodes along path $p \in P_k$
W	set of deployment configurations
I_j	set of breakpoints in the PWL function of deployment configuration $j \in W$
Γ_k	set of length-bounded cuts for OD pair k
Q	set of nodes in a length-bounded cut
Q_k^{min}	set of nodes in a minimal length-bounded cut for OD pair k
$\overrightarrow{Q}_k^{min}$	set of nodes in a forward minimal length-bounded cut for OD pair k
$\overleftarrow{Q}_k^{min}$	set of nodes in a backward minimal length-bounded cut for OD pair k
$\Lambda(\overline{\mathbf{y}})$	set of available nodes given location vector $\overline{\mathbf{y}}$
$\Phi(\overline{\mathbf{y}})$	set of covered OD pairs given location vector $\overline{\mathbf{y}}$
$\Psi_k(\overline{\mathbf{y}})$	set of all available paths for OD pair k given location vector $\overline{\mathbf{y}}$
Parameter	**Description**
c_j	deployment cost of configuration $j \in W$
n_j	number of chargers in deployment configuration $j \in W$
e	exponential service rate of each charger
q_{max}	maximum queuing time from truncating the $M/M/c$ queuing function
(U_j^l, J_j^l)	queuing time and EV flow rate of l-th breakpoint in the PWL function for configuration j
ρ_j^l	slope of l-th segment in the PWL function for configuration j
μ_j^l	length of the EV flow rate of l-th segment in the PWL function for configuration j
f_k	stochastic stream of EV flow rate of OD pair k
(s_k, t_k)	dummy nodes for the origin and destination of OD pair k
H	total EV flow volume passing through the network
B	a given budget
R	EV range
v	EV speed
ξ	constant charging time per charging stop
τ_T	time deviation level
λ_k^{min}	the minimum time required to traverse the shortest-distance path with the fewest charging stops, excluding queuing time for OD pair k
λ_k	maximum allowable total time (sum of travel time, charging time, queuing time) for OD pair k
ϵ	allowable suboptimality gap in the path disutility
a_{pi}	1 if selected path $p \in P_k$ visits node i; 0 otherwise
d_p	travel distance of path $p \in P_k$
Variable	**Description**
w_{ij}	1 if deployment configuration j is installed at node i; 0 otherwise
y_i	1 if charging facility is open at node i; 0 otherwise
z_k	1 if OD pair k is covered; 0 otherwise
σ_{ij}^l	1 if l-th segment of PWL function for configuration j at node i is activated; 0 otherwise
b_{ij}^l	proportion of activation of l-th segment of PWL function for configuration j at node i
Δ_i	EV flow rate passing through node i
q_i	expected queuing time at node i
u_p	disutility (sum of travel time, charging time and queuing time) of path p

Table A2: Model-specific parameters and decision variables in the Arc-Flow and Path-Based formulations

Model	Parameter	Description
SLM-A	δ_{ij}	shortest distance from i to j in G
	Variable	**Description**
	x_{ijk}	1 if arc (i,j) in G_k is on the selected (realized) path; 0 otherwise
	$\hat{x}_{ijk}$	1 if arc (i,j) in G_k is on the minimum-disutility reference path; 0 otherwise
	π_{ik}	dual variables associated with follower's constraints (20)
	γ_{ik}	dual variables associated with follower's constraints (21)
	θ_k	dual variables associated with follower's constraints (22)
	$\hat{q}_{ijk}$	auxiliary variables for linearizing $x_{ijk}q_j$
	$\hat{\gamma}_{ik}$	auxiliary variables for linearizing $y_i\gamma_{ik}$
Model	**Parameter**	**Description**
SLM-P	t_p	sum of travel time and charging time along path p
	Variable	**Description**
	x_p	1 if path p is the selected (realized) path; 0 otherwise
	$\hat{x}_p$	1 if path p is the minimum-disutility path; 0 otherwise
	π_k	dual variables associated with follower's constraints (32)
	γ_{kpi}	dual variables associated with follower's constraints (33)
	θ_k	dual variables associated with follower's constraints (29)
	$\hat{q}_p$	auxiliary variables for linearizing $x_p \sum_{i \in N_p} q_i$
	$\hat{\gamma}_{kpi}$	auxiliary variables for linearizing $y_i\gamma_{kpi}$

Appendix B Three road networks



(a) N25 network **(b) California highway network (CA339)**

Figure B1: Small-scale (left) and large-scale networks (right)

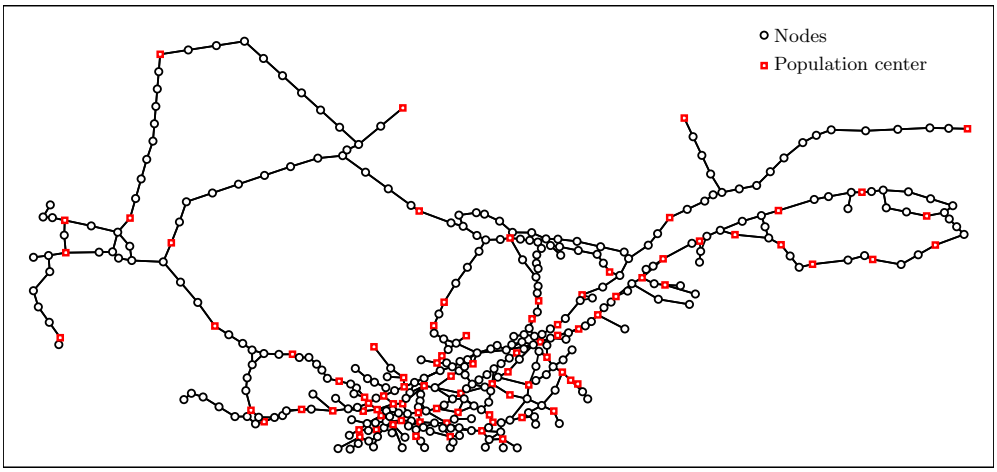


Figure B2: Québec road network (QC405) with 88 OD nodes, 405 candidate nodes, and 1003 arcs

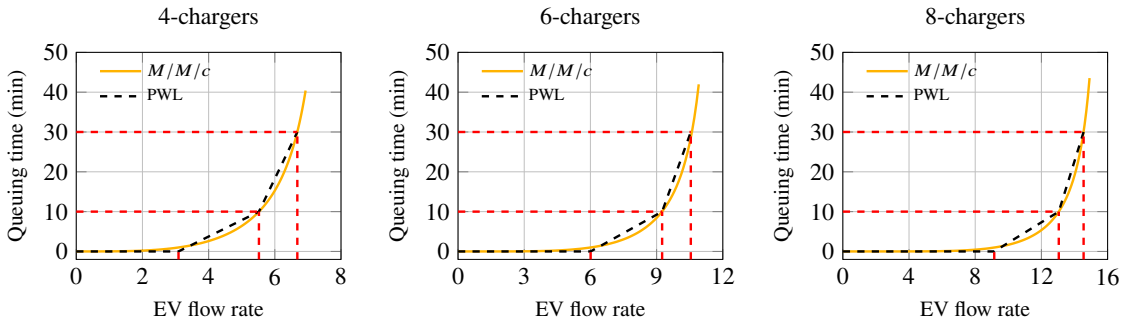


Figure B3: PWL of the $M/M/c$ queuing function under deployment configurations for the N25 Network

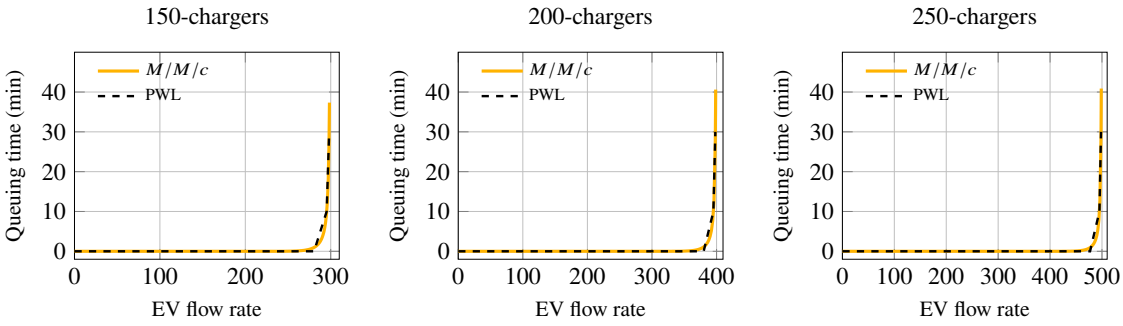


Figure B4: PWL of the $M/M/c$ queuing function under deployment configurations for the CA339 Network

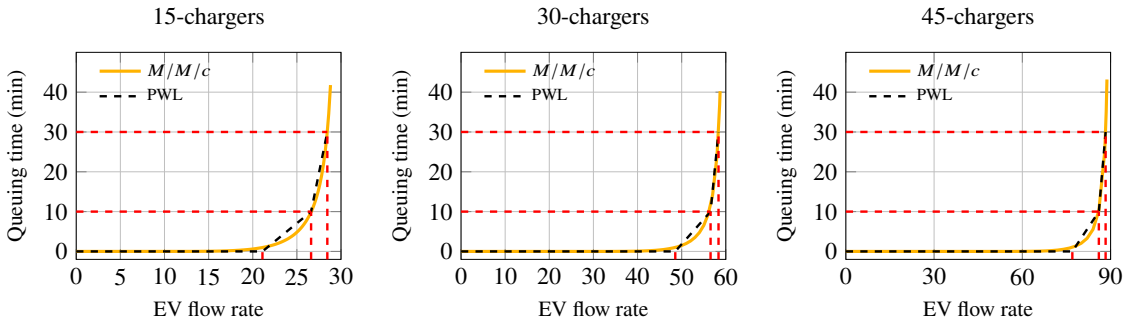


Figure B5: PWL of the $M/M/c$ queuing function under deployment configurations for the QC405 Network

Appendix C The detailed results of N25 instances

Table C1: Computational results of the proposed model and algorithm on the N25 with $R = 150$ km

τ_T	B	Obj	SLM-A	SLM-P	DA		Deploy	Max Dev	Avg Dev	Max q	Avg q
	(Mil $)	(%)	(sec)	(sec)	(sec)	#Iter				(minutes)	(minutes)
0%	0.9	23.99	2.58	0.75	0.25	3	2/8	0.00%	0.00%	0.00	0.00
	1.8	45.82	2.45	0.98	2.36	11	4/16	0.00%	0.00%	0.00	0.00
	2.7	64.17	4.02	2.39	8.50	36	6/24	0.00%	0.00%	0.00	0.00
	3.6	77.84	3.53	3.61	32.59	109	8/32	0.00%	0.00%	0.00	0.00
	4.5	88.03	3.96	2.82	22.33	107	9/40	0.00%	0.00%	0.00	0.00
10%	0.9	26.32	6.42	1.84	0.06	1	2/8	2.08%	1.22%	1.78	0.89
	1.8	53.74	5.09	1.45	0.57	3	4/16	9.58%	2.31%	5.11	1.63
	2.7	72.52	5.42	1.07	0.57	2	6/24	9.68%	2.37%	5.09	1.36
	3.6	85.48	10.40	0.86	0.70	3	8/32	9.91%	1.70%	5.08	1.26
	4.5	95.79	3.57	0.83	0.20	1	10/40	8.33%	1.14%	4.81	0.73
25%	0.9	28.72	44.89	8.13	1.11	4	2/8	23.91%	3.41%	0.20	0.10
	1.8	62.62	106.89	40.81	1.91	8	4/16	23.79%	7.77%	8.63	4.96
	2.7	79.27	194.53	67.17	34.48	66	6/24	24.60%	6.44%	7.58	3.12
	3.6	92.61	331.10	41.00	20.65	25	8/32	25.00%	5.01%	6.84	2.08
	4.5	98.04	70.25	7.74	1.29	2	10/40	25.00%	3.44%	6.00	0.96
50%	0.9	34.03	120.33	20.97	0.92	2	2/8	46.33%	13.39%	7.67	3.84
	1.8	73.37	296.51	340.68	5.39	11	4/16	47.06%	14.82%	9.89	7.20
	2.7	92.17	716.45	248.60	21.91	15	6/24	49.85%	18.75%	11.06	6.32
	3.6	99.27	1860.54	302.44	15.68	7	8/32	48.33%	10.53%	12.66	3.36
	4.5	99.95	1272.68	113.16	28.51	12	10/40	49.81%	11.53%	9.63	2.82
Avg			253.08	60.36	10.00	21.40					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Table C2: Computational results of the proposed model and algorithm on the N25 with $R = 200$ km

τ_T	B	Obj	SLM-A	SLM-P	DA		Deploy	Max Dev	Avg Dev	Max q	Avg q
	(Mil $)	(%)	(sec)	(sec)	(sec)	#Iter				(minutes)	(minutes)
	0.9	44.11	2.63	0.75	2.81	22	1/8	0.00%	0.00%	0.00	0.00
	1.8	69.89	1.89	0.58	12.70	87	3/16	0.00%	0.00%	0.00	0.00
0%	2.7	88.45	1.51	0.45	6.17	54	5/24	0.00%	0.00%	0.00	0.00
	3.6	95.16	1.67	0.49	8.84	60	7/32	0.00%	0.00%	0.00	0.00
	4.5	99.26	1.28	0.62	0.49	6	9/40	0.00%	0.00%	0.00	0.00
	0.9	48.54	12.49	5.41	1.29	13	2/8	7.46%	2.80%	7.83	3.92
	1.8	78.22	49.12	6.36	9.78	40	3/18	10.00%	2.48%	0.00	0.00
10%	2.7	91.82	123.10	6.87	17.86	61	5/26	10.00%	1.84%	0.00	0.00
	3.6	98.21	96.31	5.70	4.10	9	8/32	10.00%	1.25%	5.83	0.73
	4.5	99.92	4.11	0.63	0.21	1	10/40	10.00%	0.56%	0.50	0.05
	0.9	59.74	23.67	14.86	1.34	8	2/8	22.98%	11.82%	24.12	12.55
	1.8	90.07	69.10	30.08	8.66	28	4/16	23.70%	7.77%	13.04	5.53
25%	2.7	99.28	164.14	34.01	18.10	50	5/24	25.00%	4.60%	0.00	0.00
	3.6	100.00	163.96	3.49	0.26	1	7/32	22.41%	2.07%	0.00	0.00
	4.5	100.00	106.21	2.88	0.20	1	7/44	22.22%	0.96%	0.00	0.00
	0.9	64.16	131.90	42.06	1.77	10	2/8	46.64%	17.03%	25.97	14.85
	1.8	97.26	481.65	150.14	18.67	41	4/16	49.58%	14.67%	16.21	8.76
50%	2.7	99.95	468.31	242.62	10.68	20	6/24	48.58%	11.70%	9.57	3.09
	3.6	100.00	162.30	31.03	0.57	2	7/34	41.58%	8.22%	2.50	0.36
	4.5	100.00	214.05	24.57	0.35	1	8/44	50.00%	8.25%	0.00	0.00
Avg			113.97	30.18	6.24	25.75					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Table C3: Computational results of the proposed model and algorithm on the N25 with $R = 250$ km

τ_T	B	Obj	SLM-A	SLM-P	DA		Deploy	Max Dev	Avg Dev	Max q	Avg q
	(Mil $)	(%)	(sec)	(sec)	(sec)	#Iter				(minutes)	(minutes)
	0.9	48.64	2.18	0.81	0.39	7	2/8	0.00%	0.00%	0.00	0.00
	1.8	79.07	1.99	0.45	1.46	15	4/16	0.00%	0.00%	0.00	0.00
0%	2.7	91.27	1.12	1.42	1.42	14	6/24	0.00%	0.00%	0.00	0.00
	3.6	98.65	0.86	0.13	0.04	1	8/32	0.00%	0.00%	0.00	0.00
	4.5	99.59	0.84	0.13	0.05	1	10/40	0.00%	0.00%	0.00	0.00
	0.9	64.12	8.32	2.04	0.12	3	2/8	10.00%	5.12%	9.41	4.71
	1.8	87.75	5.68	0.78	0.06	1	4/16	10.00%	2.47%	6.52	1.63
10%	2.7	95.52	2.85	0.52	0.09	1	6/24	10.00%	1.44%	1.52	0.25
	3.6	99.25	4.86	0.50	0.13	1	8/32	10.00%	0.24%	0.00	0.00
	4.5	100.00	3.97	0.63	0.16	1	10/40	5.88%	0.15%	0.00	0.00
	0.9	72.51	20.12	29.67	1.75	7	2/8	24.37%	9.28%	15.32	7.66
	1.8	96.93	56.67	10.35	1.83	9	4/16	23.87%	5.27%	7.50	1.88
25%	2.7	100.00	18.46	2.17	0.20	1	6/24	25.00%	2.76%	0.00	0.00
	3.6	100.00	22.03	2.10	0.21	1	7/32	25.00%	2.52%	0.00	0.00
	4.5	100.00	45.78	1.99	0.19	1	7/40	18.18%	1.76%	0.00	0.00
	0.9	82.98	49.29	62.96	0.68	5	2/8	49.52%	19.85%	25.08	14.11
	1.8	99.59	254.08	105.30	0.47	2	4/16	50.00%	7.85%	22.43	5.61
50%	2.7	100.00	86.98	6.78	0.18	1	5/24	49.74%	5.54%	0.00	0.00
	3.6	100.00	94.60	9.68	0.17	1	5/36	47.06%	5.86%	0.00	0.00
	4.5	100.00	62.03	7.50	0.26	1	6/48	47.62%	4.67%	0.00	0.00
Avg			37.14	12.30	0.49	3.70					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Table C4: The objective value (%) on the N25 under different ϵ setting

τ_T	B (Mil $)	R=150 km		R=200 km		R=250 km	
		ϵ=5 min	ϵ=0 min	ϵ=5 min	ϵ=0 min	ϵ=5 min	ϵ=0 min
0	0.9	23.99	23.99	44.11	44.11	48.64	48.64
	1.8	45.82	45.82	69.89	69.89	79.07	79.07
	2.7	64.17	64.17	88.45	88.45	91.27	91.27
	3.6	77.84	77.84	95.16	95.16	98.65	98.65
	4.5	88.03	88.03	99.26	99.26	99.59	99.59
0.1	0.9	26.32	26.32	48.54	48.54	64.12	64.12
	1.8	53.74	53.74	78.22	78.22	87.75	87.75
	2.7	72.52	72.52	91.82	91.82	95.52	95.52
	3.6	85.48	85.48	98.21	98.21	99.25	99.25
	4.5	95.79	95.79	99.92	99.92	100.00	100.00
0.25	0.9	28.72	**28.45**	59.74	59.74	72.51	72.51
	1.8	62.62	62.62	90.07	90.07	96.93	96.93
	2.7	79.27	79.27	99.28	99.28	100.00	100.00
	3.6	92.61	**90.71**	100.00	100.00	100.00	100.00
	4.5	98.04	98.04	100.00	100.00	100.00	100.00
0.5	0.9	34.03	34.03	64.16	64.16	82.98	82.98
	1.8	73.37	**72.69**	97.26	**95.92**	99.59	99.59
	2.7	92.17	**90.11**	99.95	99.95	100.00	100.00
	3.6	99.27	99.27	100.00	100.00	100.00	100.00
	4.5	99.95	**99.79**	100.00	100.00	100.00	100.00

Appendix D The detailed results of CA339 instances

Table D1: Computational results of the DA algorithm on the CA339 with $R = 200$ km

τ_T	B (Mil $)	OBJ %	Opt Gap %	Time (s)	#Iter	Deploy	Max dev %	Avg dev %	Max q (minutes)	Avg q (minutes)
0%	24	61.57	0.00	1.49	4	2/300	0.00	0.00	0.00	0.00
	36	79.09	0.00	0.35	1	3/450	0.00	0.00	0.00	0.00
	48	83.53	0.00	0.38	1	4/600	0.00	0.00	0.00	0.00
	60	86.82	0.00	0.51	1	5/750	0.00	0.00	0.00	0.00
	72	89.89	0.00	0.57	1	6/900	0.00	0.00	0.00	0.00
10%	24	67.61	0.00	41.55	33	2/300	9.75	6.01	7.66	5.63
	36	85.79	0.00	6.83	7	3/450	9.97	2.05	1.95	0.65
	48	89.10	0.00	71.12	27	4/600	9.67	1.94	2.08	0.52
	60	92.04	0.00	113.42	36	5/750	9.79	1.45	0.00	0.00
	72	94.84	0.00	1.97	1	6/900	9.99	2.39	0.00	0.00
25%	24	69.55	0.00	1874.63	315	2/300	24.61	9.66	8.91	4.45
	36	90.75	0.00	4165.96	495	3/450	24.58	5.32	8.86	2.95
	48	94.93	0.26	7200.39	761	4/600	24.92	6.05	9.07	2.27
	60	96.79	1.00	7200.68	806	4/700	24.98	6.56	0.00	0.00
	72	98.50	0.93	7209.65	912	6/900	24.89	4.33	0.54	0.09
50%	24	70.41	7.19	7215.23	518	2/300	50.00	20.10	26.05	14.70
	36	91.27	2.74	7200.23	383	2/450	49.44	5.99	0.00	0.00
	48	95.05	3.25	7202.42	440	4/600	49.95	20.79	22.79	9.22
	60	99.24	0.60	7206.77	320	5/750	49.97	19.21	26.75	8.16
	72	99.99	0.01	15.50	1	6/900	50.00	9.14	11.85	1.98
Avg			0.80	2836.48	253.15					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Table D2: Computational results of the DA algorithm on the CA339 with $R = 250$ km

τ_T	B (Mil $)	OBJ %	Opt Gap %	Time (s)	#Iter	Deploy	Max dev %	Avg dev %	Max q (minutes)	Avg q (minutes)
0%	24	85.60	0.00	0.31	1	2/300	0.00	0.00	0.00	0.00
	36	89.34	0.00	0.31	1	3/450	0.00	0.00	0.00	0.00
	48	92.41	0.00	0.36	1	4/600	0.00	0.00	0.00	0.00
	60	95.26	0.00	0.41	1	5/750	0.00	0.00	0.00	0.00
	72	97.19	0.00	0.44	1	6/900	0.00	0.00	0.00	0.00
10%	24	90.14	0.00	1.07	1	2/300	9.87	0.85	0.00	0.00
	36	96.11	0.00	1.40	1	3/450	9.87	1.26	0.00	0.00
	48	98.02	0.00	1.52	1	4/600	9.89	1.10	0.00	0.00
	60	98.90	0.00	1.72	1	5/750	9.89	1.19	0.00	0.00
	72	99.57	0.00	1.42	1	6/900	9.89	1.02	0.00	0.00
25%	24	93.09	0.00	45.77	16	2/300	23.88	3.86	1.21	0.61
	36	99.35	0.00	38.36	10	3/450	24.23	3.78	2.87	0.96
	48	100.00	0.00	1.54	1	4/600	24.96	4.76	0.00	0.00
	60	100.00	0.00	2.84	1	4/650	24.09	4.06	0.00	0.00
	72	100.00	0.00	5.22	1	6/900	23.14	2.99	0.00	0.00
50%	24	93.24	0.00	1789.80	212	2/300	44.08	4.18	1.23	0.62
	36	100.00	0.00	9.68	6	3/450	46.67	11.00	0.38	0.13
	48	100.00	0.00	2.34	1	4/600	45.87	7.46	3.25	0.81
	60	100.00	0.00	2.76	1	5/750	48.75	13.53	26.58	5.32
	72	100.00	0.00	1.48	1	5/800	47.75	11.53	0.00	0.00
Avg			0.00	95.44	13.00					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Table D3: Computational results of the DA algorithm on the CA339 with $R = 300$ km

τ_T	B (Mil $)	OBJ %	Opt Gap %	Time (s)	#Iter	Deploy	Max dev %	Avg dev %	Max q (minutes)	Avg q (minutes)
0%	24	92.44	0.00	0.30	1	2/300	0.00	0.00	0.00	0.00
	36	95.31	0.00	0.37	1	3/450	0.00	0.00	0.00	0.00
	48	97.15	0.00	0.33	1	4/600	0.00	0.00	0.00	0.00
	60	98.52	0.00	0.34	1	5/750	0.00	0.00	0.00	0.00
	72	99.42	0.00	0.37	1	6/900	0.00	0.00	0.00	0.00
10%	24	95.81	0.00	159.69	106	2/300	9.89	1.17	0.00	0.00
	36	99.44	0.00	1.29	1	3/450	9.89	1.26	0.00	0.00
	48	99.83	0.00	1.21	1	4/600	9.89	1.17	0.00	0.00
	60	99.91	0.00	1.23	1	5/750	9.89	1.48	0.00	0.00
	72	99.95	0.00	1.72	1	6/900	9.88	2.53	0.00	0.00
25%	24	95.99	0.00	4349.53	635	2/300	14.85	1.63	0.00	0.00
	36	99.97	0.00	2784.62	1161	3/450	24.75	6.29	1.87	0.62
	48	100.00	0.00	4.69	2	3/500	21.94	3.00	0.00	0.00
	60	100.00	0.00	1.20	1	4/700	21.94	2.55	0.00	0.00
	72	100.00	0.00	1.00	1	4/650	17.42	1.38	0.00	0.00
50%	24	94.49	2.46	7203.19	702	1/200	17.98	0.77	0.00	0.00
	36	100.00	0.00	11.43	6	3/450	49.76	14.85	0.00	0.00
	48	100.00	0.00	4.36	1	3/600	49.01	17.59	0.00	0.00
	60	100.00	0.00	3.80	1	5/750	45.48	13.50	0.00	0.00
	72	100.00	0.00	2.72	1	5/750	49.93	14.62	0.00	0.00
Avg			0.12	726.67	131.30					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Appendix E The Detailed Results of QC405 Instances

Table E1: Computational results of the DA algorithm on the QC405 with $R = 200$ km

B (Mil $)	OBJ %	Opt Gap %	Time (s)	#Iter	Deploy	Max dev %	Avg dev %	Max q (minutes)	Avg q (minutes)
20	50.65	3.10%	7205.66	577	14/225	0.00%	0.00%	0.00	0.00
30	70.82	1.90%	7221.71	528	22/345	0.00%	0.00%	0.00	0.00
40	85.55	0.00%	385.19	92	31/465	0.00%	0.00%	0.00	0.00
50	91.57	0.00%	1.79	1	39/585	0.00%	0.00%	0.00	0.00
60	94.14	0.00%	3.01	1	47/705	0.00%	0.00%	0.00	0.00
20	66.65	12.13%	7375.33	20	13/240	10.00%	2.68%	4.95	0.71
30	86.39	4.10%	7276.08	12	20/360	10.00%	1.77%	0.25	0.01
40	94.80	0.59%	7373.66	15	30/465	10.00%	1.57%	1.15	0.05
50	97.13	0.43%	7340.58	26	37/585	10.00%	1.20%	0.00	0.00
60	98.81	0.00%	67.14	1	47/705	9.91%	1.01%	0.00	0.00
20	74.45	22.45%	7365.40	12	11/240	24.97%	9.18%	10.00	2.59
30	89.36	9.88%	7345.04	12	17/360	25.01%	6.28%	1.60	0.10
40	97.73	2.06%	7315.38	12	26/480	24.79%	6.11%	0.99	0.05
50	99.98	0.02%	7333.02	13	34/585	24.95%	4.31%	4.56	0.14
60	100.00	0.00%	4073.97	10	44/705	24.97%	4.67%	3.49	0.21
20	79.65	20.53%	7581.81	12	10/240	49.99%	13.31%	20.66	6.50
30	92.75	7.53%	7588.55	12	15/360	50.00%	18.70%	9.89	2.72
40	99.71	0.29%	7375.38	12	24/480	49.96%	15.00%	19.97	1.61
50	99.98	0.02%	7265.79	12	32/600	49.98%	11.16%	4.19	0.25
60	100.00	0.00%	1206.41	2	40/720	50.01%	14.27%	12.11	0.56
		4.25%	5435.04	69.10					

Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Table E2: Computational results of the DA algorithm on the QC405 with $R = 300$ km

τ_T	B (Mil $)	OBJ %	Opt Gap %	Time (s)	#Iter	Deploy	Max dev %	Avg dev %	Max q (minutes)	Avg q (minutes)
	20	74.03	0.00%	1.57	1	15/225	0.00%	0.00%	0.00	0.00
	30	85.74	0.00%	1.48	1	23/345	0.00%	0.00%	0.00	0.00
0%	40	91.70	0.00%	2.61	1	31/465	0.00%	0.00%	0.00	0.00
	50	95.27	0.00%	1.97	1	39/585	0.00%	0.00%	0.00	0.00
	60	97.57	0.00%	1.77	1	47/705	0.00%	0.00%	0.00	0.00
	20	85.15	0.62%	7450.22	25	15/225	10.00%	2.92%	9.91	1.98
	30	93.84	0.00%	7.76	1	23/345	9.99%	1.13%	3.63	0.18
10%	40	97.16	0.00%	4.97	1	31/465	9.89%	0.71%	0.00	0.00
	50	98.75	0.00%	8.95	1	39/585	9.89%	0.53%	0.00	0.00
	60	99.63	0.00%	22.93	1	47/705	9.89%	0.42%	0.00	0.00
	20	91.36	7.34%	7401.67	23	12/240	25.01%	8.58%	21.65	3.85
	30	99.52	0.46%	7223.22	43	20/360	24.98%	5.01%	14.43	0.83
25%	40	100.00	0.00%	533.06	4	29/465	24.76%	5.86%	12.71	0.68
	50	100.00	0.00%	27.93	1	36/570	23.99%	4.93%	0.00	0.00
	60	100.00	0.00%	19.88	1	44/690	24.95%	5.29%	0.00	0.00
	20	94.51	5.48%	7348.03	25	10/240	49.99%	12.81%	10.00	3.55
	30	99.88	0.12%	7470.43	22	18/360	49.68%	10.73%	6.27	0.56
50%	40	100.00	0.00%	864.85	2	27/465	49.95%	10.24%	7.64	0.68
	50	100.00	0.00%	107.41	1	37/585	49.85%	13.22%	2.02	0.10
	60	100.00	0.00%	62.35	1	39/720	49.85%	12.43%	0.00	0.00
Avg			0.70%	1928.15	7.85					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Table E3: Computational results of the DA algorithm on the QC405 with $R = 400$ km

τ_T	B (Mil $)	OBJ %	Opt Gap %	Time (s)	#Iter	Deploy	Max dev %	Avg dev %	Max q (minutes)	Avg q (minutes)
0%	20	86.97	0.00%	231.87	122	15/225	0.00%	0.00%	0.00	0.00
	30	95.12	0.00%	0.89	1	23/345	0.00%	0.00%	0.00	0.00
	40	98.22	0.00%	1.22	1	31/465	0.00%	0.00%	0.00	0.00
	50	99.62	0.00%	1.62	1	39/585	0.00%	0.00%	0.00	0.00
	60	99.99	0.00%	2.05	1	47/705	0.00%	0.00%	0.00	0.00
10%	20	94.98	0.68%	7219.80	247	14/225	10.01%	1.00%	0.00	0.00
	30	98.88	0.14%	7204.18	471	23/345	9.95%	0.85%	1.11	0.05
	40	99.97	0.00%	5.58	1	31/465	9.96%	0.67%	0.00	0.00
	50	100.00	0.00%	5.11	1	37/585	9.99%	0.97%	0.00	0.00
	60	100.00	0.00%	6.71	1	41/630	9.94%	0.75%	0.00	0.00
25%	20	99.14	0.66%	7332.91	124	12/240	24.88%	4.38%	3.53	0.29
	30	100.00	0.00%	1104.67	17	20/360	24.71%	4.20%	2.27	0.12
	40	100.00	0.00%	36.50	5	24/435	24.48%	4.89%	0.00	0.00
	50	100.00	0.00%	71.62	20	32/540	24.79%	5.97%	0.00	0.00
	60	100.00	0.00%	81.67	20	32/540	24.99%	6.19%	0.00	0.00
50%	20	99.90	0.10%	7250.94	158	12/240	49.60%	10.50%	29.49	3.95
	30	100.00	0.00%	1207.21	13	20/360	46.37%	8.29%	6.31	0.32
	40	100.00	0.00%	541.84	6	28/465	49.98%	13.68%	6.89	0.25
	50	100.00	0.00%	12.14	1	34/585	49.56%	13.48%	0.00	0.00
	60	100.00	0.00%	10.47	1	34/720	49.56%	13.47%	0.00	0.00
Avg			0.08%	1616.45	60.60					

Deploy: number of open nodes/number of chargers
Max q: maximum queuing time among open nodes
Avg q: average queuing time among open nodes

Appendix F Figures

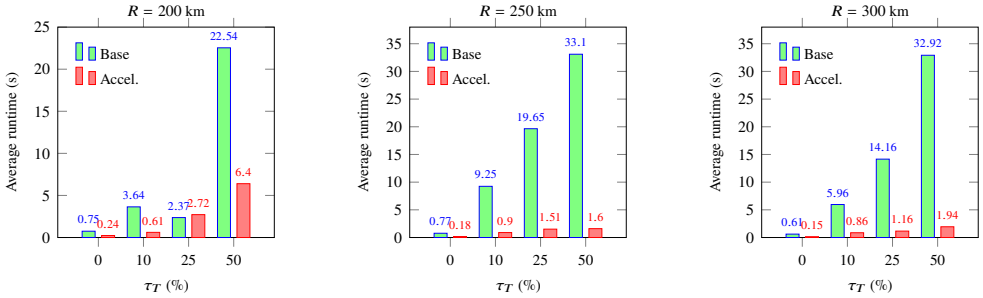


Figure F1: The average runtime of the MP per iteration of CA339 instance with and without using acceleration strategies

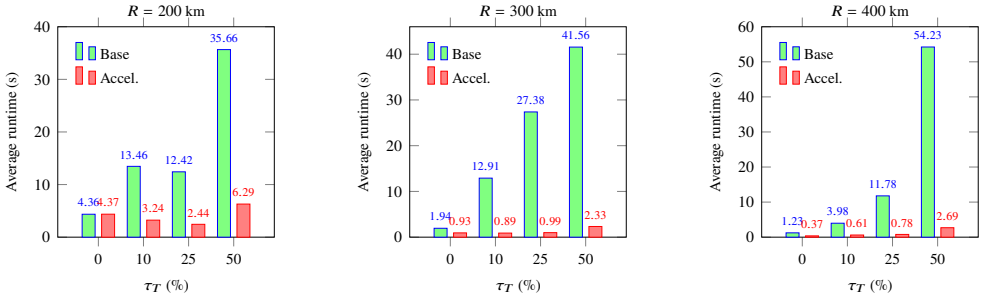


Figure F2: The average runtime of the MP per iteration of QC405 instance with and without using acceleration strategies

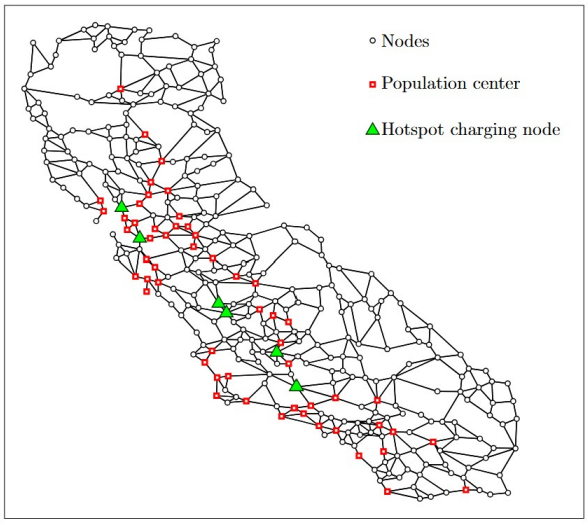


Figure F3: Hotspot charging nodes (top six) in the CA339 network

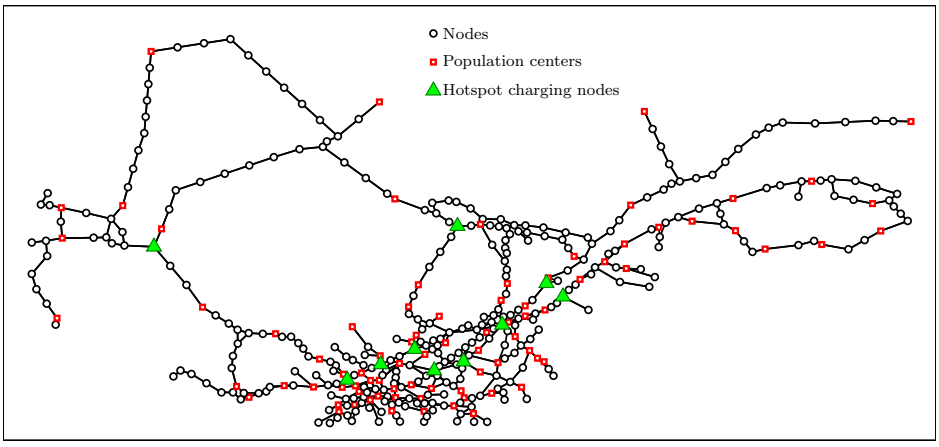


Figure F4: Hotspot charging nodes (top ten) in the QC405 network

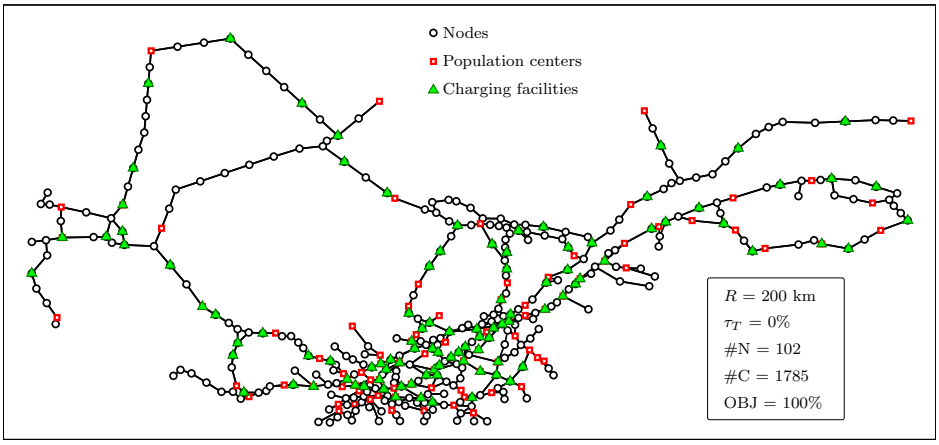


Figure F5: Deployment solution under $R = 200$ km, and full coverage OD pairs

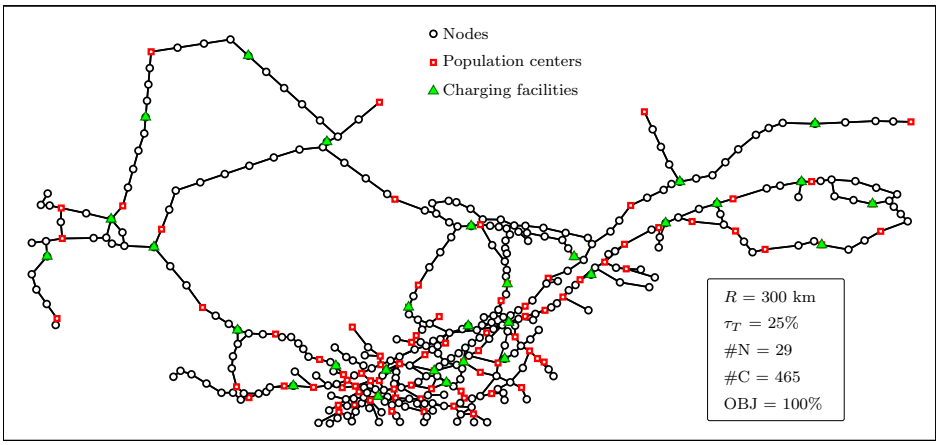


Figure F6: Deployment solution under $R = 300$ km, and full coverage OD pairs

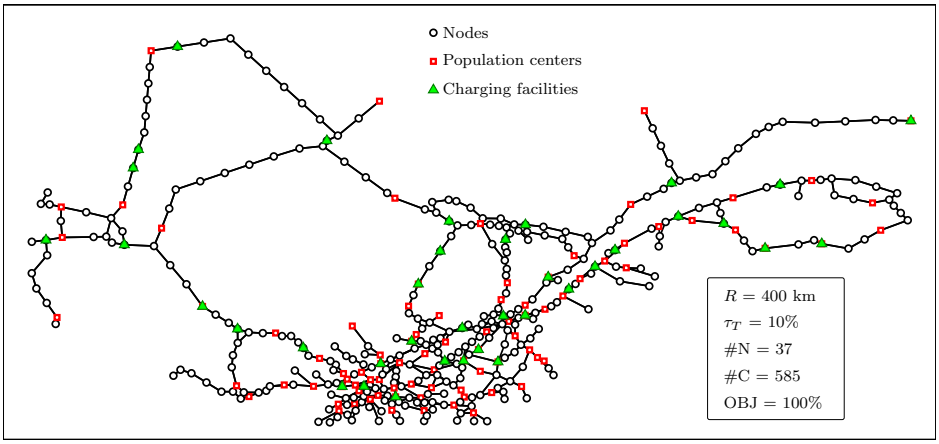


Figure F7: Deployment solution under $R = 400$ km, and full coverage OD pairs

Appendix G Adapting the DA framework to solve a related bilevel location problem

Our DA framework is a general solution approach for bilevel location problems. Its MP is responsible for generating high-quality location decisions, while its SP evaluates each generated location decision. To demonstrate the generality of the DA framework, we apply it to solve a related bilevel location problem of Kınay, Gzara, and Alumur (2023).

Note that the overall algorithmic structure of Kınay, Gzara, and Alumur (2023) is consistent with ours, in the sense that it also alternates between an MP and an SP. However, their problem setting is not identical to ours. Therefore, the DA algorithm must be adapted accordingly. The main modifications are as follows.

(a) Objective function. Kınay, Gzara, and Alumur (2023) aim to cover all OD pairs while minimizing the total deployment cost of stations and chargers. Accordingly, we modify the MP and formulate it as follows.

$$\text{[MP]} \quad \min \quad \sum_{i \in N} y_i \tag{94}$$

$$\text{s.t.} \quad 1 \leq \sum_{i \in Q} y_i \qquad \forall k \in K, Q \in \Gamma_k \tag{95}$$

$$y_i \in \{0,1\} \qquad \forall i \in N \tag{96}$$

$$+ \quad \text{logic-based cuts from } \textbf{SP} \tag{97}$$

As described in Section 6.2, constraints (95) are exponentially many. Therefore, we generate them dynamically within a branch-and-cut framework. We also employ the acceleration strategies introduced in Section 6.2.1 to further improve the computational efficiency of solving the MP.

(b) Queuing function. Although both Kınay, Gzara, and Alumur (2023) and this paper model congestion using an $M/M/c$ queuing system, they do not explicitly calculate the queuing time. Instead, they impose the following service-level requirement, ensuring that the queuing time at each station satisfies a prescribed threshold.

$$n_i = \begin{cases} 0 & \text{if} \quad \Delta_i = 0 \\ 1 & \text{if } \epsilon \leq \Delta_i \leq J^1 \\ 2 & \text{if } J^1 \leq \Delta_i \leq J^2 \\ 3 & \text{if } J^2 \leq \Delta_i \leq J^3 \\ \dots & \dots \end{cases} \qquad \mathbb{P}(q_i \leq \alpha) \geq \beta \tag{98}$$

Let n_i denote the number of chargers that must be deployed at node i when the EV flow is Δ_i, in order to ensure that the station's queuing time satisfies the prescribed service-level requirement. Parameters α and β denote the queuing-time threshold (in minutes) and the associated probability level, respectively. Unlike our setting, where each open node has a limited number of discrete deployment options, Kınay, Gzara, and Alumur (2023) allow an arbitrary number of chargers to be installed at each opened station, as long as the resulting queuing time satisfies the required service level. We also adopt this configuration to ensure consistency.

(c) EV path selection logic. In the setting of Kınay, Gzara, and Alumur (2023), for a given leader location decision, EV drivers in the lower level always travel on the shortest-distance paths under cooperative response, i.e., they do not consider charging stops or queuing time at stations when selecting paths. To remain consistent, we directly employ their SP, which solves the associated sizing problem. The corresponding logic-based cuts are revised as no-good cuts, consistent with Kınay, Gzara, and Alumur (2023).

(d) Benchmark instances. Kınay, Gzara, and Alumur (2023) conducted experiments on two networks, namely the *CA339* network and the *USEast420* network. They fixed $\beta = 90\%$ and, for each network, considered multiple combinations of R, distance-deviation levels, and α values. A time limit of one hour was imposed for each instance. In total, 32 instances were generated for each network, resulting in 64 instances overall. We apply the adapted DA algorithm to solve the same set of 64 instances under the same one-hour time limit.

(e) Experimental environment. We thank Dr. Ömer Burak Kınay for providing their source code of the proposed logic-based Benders algorithm (LBBA), which enables a fair algorithmic comparison. However, when running their code on our computational platform, we observed that the results obtained on most instances are substantially worse than the results reported in Kınay, Gzara, and Alumur (2023). This discrepancy is mainly due to the fact that our CPU configuration is less powerful than the one used in their experiments. Therefore, when reporting the performance of LBBA, we directly use the results reported in Kınay, Gzara, and Alumur (2023). The detailed comparison results are reported in Tables G1 and G2.

Table G1: Comparison of DA Algorithm and Kınay, Gzara, and Alumur (2023)'s LBBA on the USEast420 Network

R (km)	Dev	α (min)	LBBA Obj (Mil $)	LBBA # Iter	DA Obj (Mil $)	DA Gap	DA # Charger	DA # Station	DA # Iter
600	0%	30	52.2	2126	51.075	2.20%	571	55	4662
		10	57.225	2119	56.325	1.60%	641	55	4771
		5	60.225	2129	59.1	1.90%	678	55	4780
		1	63.3	2120	62.175	1.81%	719	55	4757
	10%	30	46.95	2710	46.875	0.16%	547	39	9524
		10	51.3	2705	51.075	0.44%	603	39	9504
		5	53.7	2712	53.4	0.56%	634	39	9558
		1	56.325	2705	56.325	0.00%	673	39	9485
	25%	30	44.775	1117	44.55	0.51%	532	31	7393
		10	48.525	1116	48.375	0.31%	583	31	7441
		5	50.625	1117	50.4	0.45%	610	31	7444
		1	53.55	1043	53.325	0.42%	649	31	7361
	50%	30	43.95	62	43.8	0.34%	534	25	3420
		10	47.475	58	47.175	0.64%	579	25	3801
		5	49.2	60	48.975	0.46%	603	25	3449
		1	52.35	59	51.825	1.01%	641	25	3451
400	0%	30	90.225	1826	89.025	1.35%	993	97	7160
		10	99.3	1760	98.1	1.22%	1114	97	7040
		5	104.25	1916	103.2	1.02%	1182	97	7190
		1	109.8	1903	108.525	1.17%	1253	97	7140
	10%	30	85.65	1435	84.825	0.97%	973	79	6720
		10	93.525	1450	92.925	0.65%	1081	79	6460
		5	97.5	1458	96.75	0.78%	1132	79	6530
		1	103.2	1513	102.525	0.66%	1209	79	6510
	25%	30	82.8	71	83.175	-0.45%	979	65	5210
		10	89.85	66	90.225	-0.42%	1073	65	5160
		5	93.675	71	94.125	-0.48%	1125	65	5245
		1	98.85	67	99.825	-0.98%	1201	65	5220
	50%	30	82.575	1	81.075	1.85%	979	51	4640
		10	89.025	1	87.45	1.80%	1064	51	4340
		5	92.925	1	91.05	2.06%	1112	51	4570
		1	97.875	1	95.925	2.03%	1177	51	4730
	Avg					0.81%			

LBBA: Kınay, Gzara, and Alumur (2023)

Table G2: Comparison of DA Algorithm and Kınay, Gzara, and Alumur (2023)'s LBBA on the CA339 Network

R (km)	Dev	α (min)	LBBA Obj (Mil $)	LBBA # Iter	DA Obj (Mil $)	DA Gap	DA # Charger	DA # Station	DA # Iter
250	0%	30	14.85	2175	14.475	2.59%	203	4	14359
		10	15.45	2229	15.075	2.49%	212	4	14531
		5	15.9	2233	15.525	2.42%	220	4	15553
		1	16.65	2233	16.275	2.30%	231	3	15371
	10%	30	13.95	2167	13.95	0.00%	182	3	12597
		10	14.475	2167	14.4	0.52%	189	3	12912
		5	14.85	2132	14.7	1.02%	199	3	12862
		1	15.525	2032	15.375	0.98%	261	3	13057
	25%	30	13.65	1709	13.65	0.00%	210	2	8336
		10	14.1	1767	14.1	0.00%	218	2	8030
		5	14.4	1770	14.4	0.00%	220	2	8526
		1	15.15	1710	15.075	0.50%	227	2	8397
	50%	30	13.65	1593	13.65	0.00%	195	2	6201
		10	14.1	1592	14.1	0.00%	205	2	6271
		5	14.4	1587	14.4	0.00%	215	2	6153
		1	15.15	1588	15.075	0.50%	228	2	6400
200	0%	30	25.8	2491	24.225	6.50%	309	7	12157
		10	26.925	2490	25.2	6.85%	322	7	13976
		5	27.75	2488	25.95	6.94%	332	7	11566
		1	29.1	2492	27.075	7.48%	347	7	11250
	10%	30	24.675	2446	24.6	0.30%	318	5	16499
		10	25.35	2434	25.425	-0.29%	329	5	15815
		5	25.95	2389	26.025	-0.29%	337	5	16568
		1	27.075	2270	27.3	-0.82%	354	5	16545
	25%	30	24.375	2309	23.775	2.52%	309	4	8721
		10	24.975	2231	24.6	1.52%	320	4	8637
		5	25.5	2247	25.275	0.89%	329	4	8540
		1	26.55	2257	26.4	0.57%	344	4	9125
	50%	30	23.7	1309	23.475	0.96%	307	3	9143
		10	24.3	1342	24.075	0.93%	315	3	8973
		5	24.9	1215	24.6	1.22%	322	3	9214
		1	25.875	1380	25.725	0.58%	337	3	9204
	Avg					1.54%			

LBBA: Kınay, Gzara, and Alumur (2023)

References

Anjos MF, Gendron B, Joyce-Moniz M, 2020 Increasing electric vehicle adoption through the optimal deployment of fast-charging stations for local and long-distance travel. European Journal of Operational Research 285(1):263–278.

Arslan O, Jabali O, Laporte G, 2018 Exact solution of the evasive flow capturing problem. Operations Research 66(6):1625–1640.

Arslan O, Karaşan OE, 2016 A Benders decomposition approach for the charging station location problem with plug-in hybrid electric vehicles. Transportation Research Part B: Methodological 93:670–695.

Arslan O, Karaşan OE, Mahjoub AR, Yaman H, 2019 A branch-and-cut algorithm for the alternative fuel refueling station location problem with routing. Transportation Science 53(4):1107–1125.

Besançon M, Anjos MF, Brotcorne L, 2021 Complexity of near-optimal robust versions of multilevel optimization problems. Optimization Letters 15(8):2597–2610.

Besançon M, Anjos MF, Brotcorne L, 2024 Robust bilevel optimization for near-optimal lower-level solutions. Journal of Global Optimization 90(4):813–842.

Capar I, Kuby M, Leon VJ, Tsai YJ, 2013 An arc cover–path-cover formulation and strategic analysis of alternative-fuel station locations. European Journal of Operational Research 227(1):142–151.

Chromy E, Misuth T, Kavacky M, 2011 Erlang C formula and its use in the call centers. Advances in Electrical and Electronic Engineering 9(1):7–13.

Chromy E, Misuth T, Weber A, 2012 Application of Erlang formulae in next generation networks. International Journal of Computer Network and Information Security 4(1):59–66.

Chung SH, Kwon C, 2015 Multi-period planning for electric car charging station locations: A case of Korean expressways. European Journal of Operational Research 242(2):677–687.

Dan T, Marcotte P, 2019 Competitive facility location with selfish users and queues. Operations Research 67(2):479–497.

Göpfert P, Bock S, 2019 A branch&cut approach to recharging and refueling infrastructure planning. European Journal of Operational Research 279(3):808–823.

Guillet M, Schiffer M, 2025 Coordinating charging request allocation between self-interested navigation service platforms. INFORMS Journal on Computing 37(2):293–314.

He J, Yang H, Tang TQ, Huang HJ, 2018 An optimal charging station location model with the consideration of electric vehicle's driving range. Transportation Research Part C: Emerging Technologies 86:641–654.

Huang Y, Kockelman KM, 2020 Electric vehicle charging station locations: Elastic demand, station congestion, and network equilibrium. Transportation Research Part D: Transport and Environment 78:102179.

Kınay ÖB, Gzara F, Alumur SA, 2021 Full cover charging station location problem with routing. Transportation Research Part B: Methodological 144:1–22.

Kınay ÖB, Gzara F, Alumur SA, 2023 Charging station location and sizing for electric vehicles under congestion. Transportation Science 57(6):1433–1451.

Kuby M, Lim S, 2005 The flow-refueling location problem for alternative-fuel vehicles. Socio-Economic Planning Sciences 39(2):125–145.

Lamontagne S, Carvalho M, Frejinger E, Gendron B, Anjos MF, Atallah R, 2023 Optimising electric vehicle charging station placement using advanced discrete choice models. INFORMS Journal on Computing 35(5):1195–1213.

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

Legault R, Frejinger E, 2025 A model-free approach for solving choice-based competitive facility location problems using simulation and submodularity. INFORMS Journal on Computing 37(3):603–622.

Mahmutoğulları Ö, Yaman H, 2023 Robust alternative fuel refueling station location problem with routing under decision-dependent flow uncertainty. European Journal of Operational Research 306(1):173–188.

Mahmutoğulları Ö, Yaman H, 2024 Mathematical formulations for the multi-period alternative fuel refueling station location problem with routing under decision-dependent flow dynamics. Transportation Research Part B: Methodological 186:102999.

MirHassani SA, Ebrazi R, 2013 A flexible reformulation of the refueling station location problem. Transportation Science 47(4):617–628.

Paneque MP, Bierlaire M, Gendron B, Azadeh SS, 2021 Integrating advanced discrete choice models in mixed integer linear optimization. Transportation Research Part B: Methodological 146:26–49.

Park H, Lee C, 2024 An exact algorithm for maximum electric vehicle flow coverage problem with heterogeneous chargers, nonlinear charging time and route deviations. European Journal of Operational Research 315(3):926–951.

Simchi-Levi D, Berman O, 1988 A heuristic algorithm for the traveling salesman location problem on networks. Operations Research 36(3):478–484.

Sugishita N, Sevim I, Carvalho M, Dems A, Atallah R, 2025 Fair network design problem: An application to EV charging station capacity expansion. Technical Report CIRRELT–2025–13, Université de Montréal.

Xie F, Lin Z, 2021 Integrated us nationwide corridor charging infrastructure planning for mass electrification of inter-city trips. Applied Energy 298:117142.

Xie F, Liu C, Li S, Lin Z, Huang Y, 2018 Long-term strategic planning of inter-city fast charging infrastructure for battery electric vehicles. Transportation Research Part E: Logistics and Transportation Review 109:261–276.

Xu M, Meng Q, 2020 Optimal deployment of charging stations considering path deviation and nonlinear elastic demand. Transportation Research Part B: Methodological 135:120–142.

Yıldız B, Arslan O, Karaşan OE, 2016 A branch and price approach for routing and refueling station location model. European Journal of Operational Research 248(3):815–826.

Zhang A, Kang JE, Kwon C, 2017 Incorporating demand dynamics in multi-period capacitated fast-charging location planning for electric vehicles. Transportation Research Part B: Methodological 103:5–29.

Zhang H, Moura SJ, Hu Z, Song Y, 2016 PEV fast-charging station siting and sizing on coupled transportation and power networks. IEEE Transactions on Smart Grid 9(4):2595–2605.