Electric Vehicle Routing with Public Charging Stations

Nicholas D Kullman, Justin C. Goodson, Jorge E. Mendoza

Electric Vehicle Routing with Public Charging Stations

Nicholas D. Kullman, Justin C. Goodson, and Jorge E. Mendoza

Abstract

We introduce the electric vehicle routing problem with public-private recharging strategy in which vehicles may recharge en-route at public charging infrastructure as well as at a privately-owned depot. To hedge against uncertain demand at public charging stations, we design routing policies that anticipate station queue dynamics. We leverage a decomposition to identify good routing policies, including the optimal static policy and fixed-route-based rollout policies that dynamically respond to observed queues. The decomposition also enables us to establish dual bounds, providing a measure of goodness for our routing policies. In computational experiments using real instances from industry, we show the value of our policies to be within ten percent of a dual bound. Further, we demonstrate that our policies significantly outperform the industry-standard routing strategy in which vehicle recharging generally occurs at a central depot. Our methods stand to reduce the operating costs associated with electric vehicles, facilitating the transition from internal-combustion engine vehicles.

1 Introduction

Electric vehicles (EVs) are beginning to replace internal-combustion engine vehicles (CVs) in supply chain distribution and in service routing. Logistics firms such as FedEx (2017), UPS (2018), Anheuser-Busch (2017) and La Poste (2011) are increasingly incorporating EVs into their commercial fleets, which have historically been comprised of CVs. EVs are also being adopted in home healthcare (Ferrándiz et al. 2016), utilities service (Orlando Utilities Commission 2018), and vehicle repair (Tesla 2018). Despite their increase in popularity, EVs pose operational challenges to which their CV counterparts are immune. For instance, EVs' driving ranges are often much less than that of CVs, charging infrastructure is still relatively sparse compared to the network of refueling stations for CVs, and the time required to charge an EV can range from 30 minutes to several hours – orders of magnitude longer than the time needed to refuel a CV (Pelletier et al. 2016). Companies choosing to adopt electric vehicles require fleet management tools that address these additional challenges.

The operations research community has responded with a body of work on electric vehicle routing problems (E-VRPs), an extension to the existing literature on (conventional) vehicle routing problems (VRPs). E-VRPs address many of the same variants that exist in the VRP domain, such as time-windows, restrictions on freight capacity, mixed fleet, and technician routing; for examples, see Schneider et al. (2014) and Villegas et al. (2018). Nearly all existing E-VRP research makes the assumption that the charging infrastructure utilized by the EVs is privately owned, i.e., that the EV has priority access to the charging infrastructure and may begin charging immediately when it arrives to a charging station (CS). While companies may have a depot at which this assumption holds, most do not have the means to acquire charging infrastructure outside the depot. If companies wish to use only the charging infrastructure that is privately-owned, then the EVs are restricted to charging only at the depot. We refer to this recharging strategy as *private-only* or *depot-only*.

Alternatively, we can relax the assumption of using only privately-owned CSs and consider the case where the vehicle may utilize public *extradepot* CSs – those available at locations such as municipal buildings, parking facilities, car dealerships, and grocery stores. We refer to this recharging strategy as *public-private*. At public CSs, all EVs share access to the charging terminals. If a vehicle arrives to charge and finds all terminals in use, it must wait for one to become available or seek another CS. While providing additional flexibility, the public-private strategy introduces uncertainty, which firms often wish to avoid.

Villegas et al. (2018) compares the private-only and public-private strategies in the case of French electricity provider ENEDIS who is replacing a portion of their CV fleet with EVs. Under the assumption of zero waiting times at public charging stations, they find that for the routes which cannot be serviced in a single charge, solutions using the public-private strategy offered savings up to 16% over those using private-only. Despite the suggested savings, ENEDIS chose not to implement the public-private recharging strategy, citing the uncertainty in availability at public charging infrastructure. This reduces the utility of EVs as members of the fleet, potentially impeding their broader adoption.

In an attempt to recapture this lost utility and encourage the use of the public-private recharging strategy, we provide in this work dynamic routing solutions that specifically address the uncertainty at public charging infrastructure. We demonstrate these routing solutions on real instances, using customer data from the ENEDIS instances of Villegas et al. (2018) and charging station data from the French national government. We claim the following contributions in this work:

- We introduce a new variant of the E-VRP: the E-VRP with public-private recharging strategy (E-VRP-PP), where demand at public charging stations is unknown and follows a realistic queuing process. We model the E-VRP-PP as a Markov decision process (MDP) and propose an approximate dynamic programming solution that allows the route planner to adapt to realized demand at public CSs.
- We offer a decomposition of the E-VRP-PP that facilitates the search for good policies. The decomposition allows the use of machinery from static and deterministic routing in solution methods for our stochastic and dynamic routing problem. Using this machinery, specifically a Benders-based branch-and-cut algorithm, we propose static and dynamic routing policies, including the optimal static policy and dynamic fixed-route-based lookaheads (rollouts).
- Using the same decomposition and Benders-based algorithm in conjunction with an information relaxation, we establish the value of an optimal policy with perfect information, which serves as a bound on the value of the optimal policy.
- In solving the subproblem of the Benders-based algorithm, we address a new variant of the fixed-route vehicle charging problem (FRVCP): the FRVCP with time-dependent waiting times and discrete charging decisions. In general, FRVCPs deal with the problem of ensuring energy feasibility for electric vehicle routes. We modify the labeling algorithm of Froger et al. (2019) to solve this new variant exactly. Additionally, to help bridge the gap between VRP and E-VRP research, we provide an open-source implementation of Froger et al.'s algorithm for the FRVCP.
- We demonstrate the application of our routing policies and the establishment of bounds on real world instances, which we make publicly available via the Vehicle Routing Problem Repository

(VRP-REP) (Mendoza et al. 2014). We show that our routing policies are competitive with the optimal policy, within an average of 10% for the majority of instances. We further show that all of our policies under the public-private recharging strategy soundly outperform the optimal solution under the industry-standard private-only recharging strategy, with our best policies offering savings of over 23% on average. Lastly, we show that these savings may increase substantially with the development of future technologies that grant the decision maker additional information regarding the availability of public charging stations. These results lend further motivation for companies to adopt the public-private recharging strategy, which may extend EVs' utility in commercial applications and facilitate the transition from internal-combustion engine vehicles.

In addition, we also improve on the perfect information dual bound by developing nonlinear information penalties that punish the decision maker for using information about the future to which they would not naturally have access. While our application of these penalties was limited to a small artificial instance, its success marks a first in vehicle routing, serving as a proof of concept for future research.

The remainder of the paper is organized as follows. We define the problem and formulate the dynamic routing model in §2. In §3 we review relevant EV routing literature. In §4 we explain the role of fixed routes in solving the E-VRP-PP, especially in the context of a decomposition, which we describe in the same section. We then outline our routing policies in §5 and detail the derivation of dual bounds for these policies in §6. Finally, we discuss computational experiments in §7 and provide concluding remarks in §8.

2 Problem Definition and Model

We address the *electric vehicle routing problem with public-private recharging strategy* (E-VRP-PP). The problem is characterized by making routing and charging decisions for an electric vehicle which visits a set of customers and returns to a depot from which it started. These decisions are subject to energy feasibility constraints. To ensure energy feasibility, the EV may need to stop and charge at CSs at which it may encounter a queue. The objective is to minimize the expected time to visit all customers and return to the depot, including any time spent detouring to, queuing at, and charging at CSs. We define the problem then formulate the MDP model.

2.1 Problem Definition

We have a set of known customers $\mathcal{N} = \{1, \dots, N\}$ and CSs $\mathcal{C} = \{0, N+1, \dots, N+C\}$ and a single EV. At time 0, the EV begins at the depot, which we denote by node $0 \in \mathcal{C}$. It then traverses arcs in the complete graph $\mathcal{G}$ with vertices $V = \mathcal{N} \cup \mathcal{C}$. The vehicle must visit each customer and return to the depot. We assume the time and energy required to travel between $i, j \in V$ is deterministic and known to be t_{ij} and e_{ij}, respectively. We also assume the triangle inequality holds, so for any $i, j, k \in V$, we have $t_{ik} \leq t_{ij} + t_{jk}$ and $e_{ik} \leq e_{ij} + e_{jk}$.

To make its journey energy-feasible, the EV may restore its energy at a CS $c \in \mathcal{C}$ before or after customer visits. The depot is private, meaning the EV can always access the charging terminals (or *chargers*) at the depot and may therefore begin charging immediately. In contrast, we assume extradepot CSs $\mathcal{C}' = \mathcal{C} \setminus \{0\}$ are public, so the chargers may be occupied by other EVs. We assume the EV is unaware of the demand at extradepot CSs prior to its arrival. (This represents the worst-case scenario for EV operators, as routing solutions can only improve as more information on CS demand becomes available. Access to real-time data on CS demand, while improving, is also still an exception to the norm.)

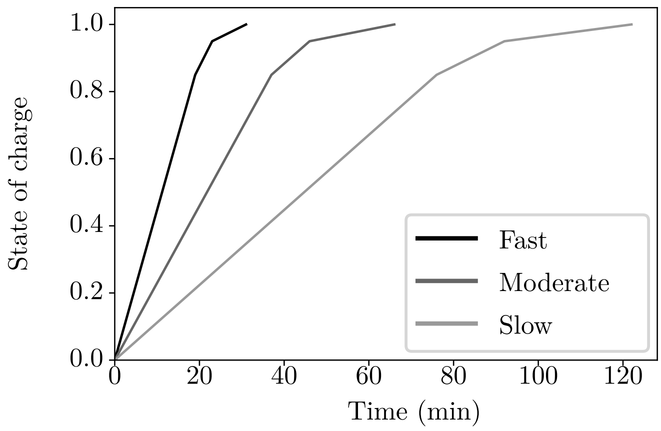

Figure 1: The vehicle's charging function for different charging technologies. We assume a concave piecewise-linear charging function as in Montoya et al. (2017).

If all chargers are occupied when the EV arrives, it must either queue or leave. We model queuing dynamics at extradepot CSs $c \in \mathcal{C}'$ as pooled first-come-first-served systems with ψ_c identical chargers, infinite capacity, and exponential inter-arrival and service times with known rate parameters ($p_{c,\text{arrive}}$ and $p_{c,\text{depart}}$, respectively): M/M/ψ_c/∞. If a vehicle queues at a CS it must remain in queue until a charger becomes available, after which it must charge. When the EV charges, it may restore its energy to a capacity $q \in \mathcal{Q}$, where $\mathcal{Q}$ is a set of discrete energy levels, such as every 10% (in which case $\mathcal{Q} = \{0, 0.1Q, \ldots, 0.9Q, Q\}$). We assume a concave piecewise-linear charging function where the EV accumulates charge faster at lower energies than at higher energies (see Figure 1). These piecewise-linear charging functions were shown in Montoya et al. (2017) to be a good approximation of actual performance. In the same study, the authors also demonstrate that the use of a simple linear approximation leads to solutions that may be either overly expensive or infeasible. We assume that the energy levels of the breakpoints in the piecewise-linear charging functions are also elements in $\mathcal{Q}$.

2.2 Model

We formulate the E-VRP-PP as an MDP whose components are defined as follows. We provide a schematic of an example epoch in Figure 2.

States.

An epoch $k \in \{0, \ldots, K\}$ is triggered when a vehicle arrives to a new location, reaches the front of the queue at a CS, or completes charging. At each epoch we describe the state of the system by the vector $s_k = \left(t_k, i_k, q_k, \bar{\mathcal{N}}_k, z_k, \mathcal{M}_k\right)$, which contains all information required for making routing and charging decisions: the current time $t_k \in \mathbb{R}_{\geq 0}$; the vehicle's current location $i_k \in V$; the energy currently in the vehicle's battery $q_k \in [0, Q]$; the set of customers that have not yet been visited $\bar{\mathcal{N}}_k \subseteq \mathcal{N}$; the vehicle's position in queue at its current location $z_k \in \mathbb{N}$ (defined to be 1 when $i_k \in \{0\} \cup \mathcal{N}$); and its memory of extradepot CS queue observations $\mathcal{M}_k$. Specifically, $\mathcal{M}_k$ is a set consisting of a memory $m = (m^c, m^l, m^t)$ for each extradepot CS $m^c \in \mathcal{C}'$ indicating the number of vehicles observed at the CS m^l and at what time m^t. If a CS c has not yet been observed, then its entry in $\mathcal{M}_k$ is $(c, \emptyset, \emptyset)$. With this definition, $\mathcal{M}_k \in \mathcal{C}' \times \mathbb{N} \times \mathbb{R}_{\geq 0}$, yielding a system state space of $\mathcal{S} = \mathbb{R}_{\geq 0} \times V \times [0, Q] \times \mathcal{N} \times \mathbb{N} \times \mathcal{C}' \times \mathbb{N} \times \mathbb{R}_{\geq 0}$.

The system is initialized in epoch 0 at time 0 with the vehicle at the depot, the battery at maximum capacity, and all customers yet to be visited:

$$s_0 = \left(0, 0, Q, \mathcal{N}, 1, \bigcup_{c \in \mathcal{C}'} (c, \emptyset, \emptyset)\right). \tag{1}$$

The problem ends at some epoch K when all customers have been visited and the EV returns to the depot: $s_K \in \{(t_K, 0, q_K, \emptyset, 1, \mathcal{M}_K) | t_K \in \mathbb{R}_{\geq 0}; q_K \in [0, Q]; \mathcal{M}_K \in \mathcal{C}' \times \mathbb{N} \times \mathbb{R}_{\geq 0}\}$.

Actions.

Given a pre-decision state s_k in some epoch k, the action space $\mathcal{A}(s_k)$ defines the possible actions that may be taken from that state. Informally, $\mathcal{A}(s_k)$ consists of energy-feasible routing and charging decisions. We define actions $a \in \mathcal{A}(s_k)$ to be location-charge pairs $a = (a^i, a^q)$ and formally define the action space as

$$\begin{aligned}
\mathcal{A}(s_k) = \Bigg\{ &(a^i, a^q) \in \{\bar{\mathcal{N}}_k \cup \mathcal{C}\} \times [0, Q] : \\
&a^i = i_k, a^q = q_k, \\
&\quad i_k \in \mathcal{C}' \wedge \psi_{i_k} < z_k && (2) \\
&a^i = i_k, a^q \in \left\{ \tilde{q} \in \mathcal{Q} \middle| \tilde{q} > q_k \wedge \right. \\
&\left. \left(\left(\exists c \in \mathcal{C}, \exists j \in \bar{\mathcal{N}}_k : \tilde{q} \geq e_{i_k j} + e_{jc} \right) \vee \left(\bar{\mathcal{N}}_k = \emptyset \wedge \tilde{q} \geq e_{i_k 0} \right) \right) \right\}, \\
&\quad i_k \in \mathcal{C} \wedge z_k \leq \psi_{i_k} && (3) \\
&a^i \in \bar{\mathcal{N}}_k, a^q = q_k - e_{i_k a^i}, \\
&\quad (\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) && (4) \\
&a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i}, \\
&\quad q_k \geq e_{i_k a^i} \Bigg\}. && (5)
\end{aligned}$$

Equation (2) defines the queuing action, in which the vehicle waits in queue until a charger becomes available. In this case, its location and charge remain unchanged. Queuing actions are feasible when the EV is at an extradepot charging station without available chargers. Equation (3) defines the charging actions. The allowable charge levels are those which are greater than the EV's current charge and which allow the EV to reach a customer and a subsequent CS (unless $\bar{\mathcal{N}}_k = \emptyset$, in which case it must charge enough to be able to reach the depot). Charging actions are present in the action space when the vehicle resides at a charging station with an available charger. Equation (4) defines routing decisions to unvisited customers. These actions are permitted so long as the vehicle has sufficient charge to reach the customer and a subsequent CS. Finally, equation (5) defines routing decisions to charging station nodes. We require that the vehicle have sufficient charge to reach the charging station. Note that we also disallow visits to CSs after a charging action, as the vehicle can recharge at at most one CS between customers (or between the depot and the first customer, or between the last customer and the depot).

Pre-to-Post-decision Transition.

Following the selection of an action $a = (a^i, a^q) \in \mathcal{A}(s_k)$ from the pre-decision state s_k, we undergo a deterministic transition to the post-decision state $s_k^a = \left(t_k^a, i_k^a, q_k^a, \bar{\mathcal{N}}_k^a, z_k^a, \mathcal{M}_k^a\right)$. In s_k^a we update the vehicle's new location and charge, which are inherited from the action: $i_k^a = a^i$, $q_k^a = a^q$. We also update the set of unvisited customers: $\bar{\mathcal{N}}_k^a = \bar{\mathcal{N}}_k \setminus \{a^i\}$. The system time, position in queue, and queue memory remain unchanged from the pre-decision state.

Information and Post-to-Pre-decision Transition.

The system transitions from a post-decision state s_k^a to a pre-decision state s_{k+1} when one of the following events occurs to trigger the next decision epoch: the vehicle reaches a new location, the vehicle reaches the front of a queue, or the vehicle completes a charging operation. At this time, we update the system time, the position in queue, and the queue memory, which were unchanged in the pre-to-post-decision transition. If arriving to a new location or reaching the front of the queue, our transition to s_{k+1} may be stochastic and depend on the observation of exogenous information. For instance, if arriving to an extradepot CS, then we observe exogenous information in the form of the queue length. If reaching the front of a CS's queue, we observe the time the vehicle waits before a charger becomes available.

We define the exogenous information observed in epoch k to be W_{k+1}, a pair consisting of a time and position in queue: $W_{k+1} = (w^t, w^z)$. The set of all exogenous information that may be observed given a post-decision state is called the information space $\mathcal{I}(s_k^a)$ and is defined as

$$
\begin{aligned}
\mathcal{I}(s_k^a) = \Big\{ & (w^t, w^z) \in \mathbb{R}_{\geq 0} \times \mathbb{N} : \\
& w^t = t_k^a + t_{i_{k-1} i_k^a}, w^z = 1, \\
& \quad i_k^a \neq i_{k-1} \wedge i_k^a \in \mathcal{N} \cup \{0\} && (6) \\
& w^t = t_k^a + \bar{u}(q_{k-1}, q_k^a), w^z = z_k^a, \\
& \quad q_k^a > q_{k-1} && (7) \\
& w^t = t_k^a + t_{i^{k-1} i_k^a}, w^z \in \mathbb{N}, \\
& \quad i_k^a \neq i_{k-1}^a \wedge i_k^a \in \mathcal{C}' && (8) \\
& w^t \in (t_k^a, \infty), w^z = \psi_{i_k^a}, \\
& \quad i_k^a = i_{k-1} \wedge q_k^a = q_{k-1} \Big\}, && (9)
\end{aligned}
$$

where $\bar{u} : [0, Q]^2 \to \mathbb{R}_{\geq 0}$ is a function defining the time required to charge from some energy level q_{initial} to another charge level q_{final} according to the vehicle's charging function. We assume a concave piecewise-linear charging function as shown in Figure 1.

Equations (6) and (7), respectively, define the (deterministic) information observed when the vehicle arrives to the depot or to a customer and when the vehicle completes a charging operation. In equation (6), the observed time is simply the previous time plus the travel time to reach the new node, and the vehicle's position is one by definition. In equation (7), we update the time to account for the time that the vehicle spent charging, and there is no update to the vehicle's position in queue, which we assume to be the same as when it began charging. The information defined by equations (8) and (9) involves uncertainty in queue dynamics at extradepot CSs $c \in \mathcal{C}'$. In equation (8), the EV has just arrived to an extradepot CS, so the time is deterministic, but we observe an unknown queue length. In equation (9), the EV has finished queuing. We assume the vehicle now occupies the last ($\psi_{i_k^a}$-th) charger, but the time of the next epoch is unknown.

Given exogenous information $W_{k+1} = (w^t, w^z)$ and post-decision state s_k^a, we transition to the pre-decision state s_{k+1} where $t_{k+1} = w^t$ and $z_{k+1} = w^z$. If we just arrived to an extradepot CS, then we

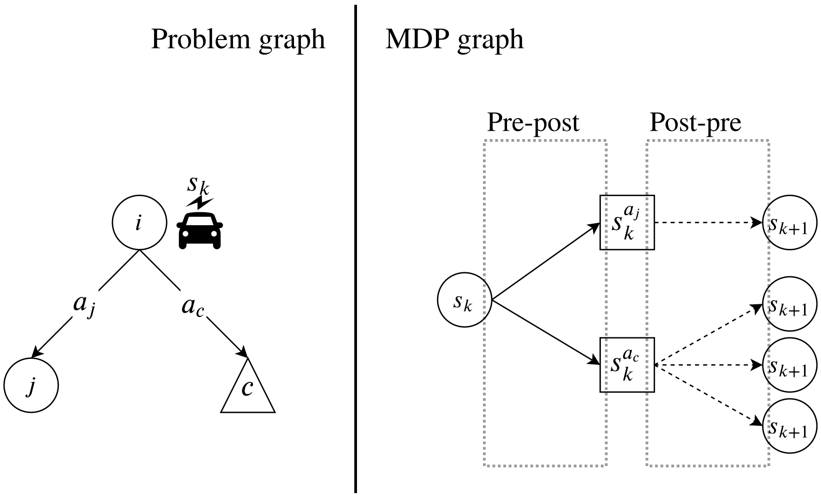

Figure 2: Depicting an example epoch. (Left) Two actions are possible from customer i in state s_k: relocating to customer j (a^j) and to CS c (a_c). (Right) After choosing an action, the pre-to-post-decision transition occurs (solid arrows): we immediately update state elements whose transitions are always deterministic; the action is not yet implemented. Then the post-to-pre-decision transition occurs (dashed arrows): the chosen action is implemented, exogenous information is observed, and the remaining state elements are updated.

also update the memory for the queue at that CS, setting it to (i_k^a, w^z, w^t). The other state components, all of which were updated in the transition to the post-decision state, remain the same.

Contribution Function.

When we select an action $a = (a^i, a^q)$ from a pre-decision state s_k, we incur cost

$$C(s_k, a) = \begin{cases} t_{i_k a^i} & a^i \neq i_k \\ \bar{u}(q_k, a^q) & a^q > q_k \\ (z_k - \psi_{i_k})/(\psi_{i_k} p_{i_k,\text{depart}}) & \text{otherwise.} \end{cases} \tag{10}$$

In equation (10), the first case corresponds to traveling to a new node, for which we incur cost equal to the travel time to reach the node. In the second case, the action is charging, and we incur cost equal to the charging time. Finally, in the third case, we have chosen to wait in queue, for which we incur cost equal to the expected waiting time conditional on the queue length.

Objective Function.

Let Π denote the set of Markovian deterministic policies, where a policy $\pi \in \Pi$ is a sequence of decision rules $(X_0^\pi, X_1^\pi, \dots, X_K^\pi)$ where each $X_k^\pi : s_k \to \mathcal{A}(s_k)$ is a function mapping a state to an action. We seek an optimal policy $\pi^\star \in \Pi$ that minimizes the expected total cost of the tour conditional on the initial state:

$$\tau(\pi^\star) = \min_{\pi\in\Pi} \mathbb{E}\left[\sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) \middle| s_0\right]. \tag{11}$$

In our solution methods, it is often convenient to think of a policy beginning from a given pre-decision state $s_{k'}$. In this case, a policy is defined as a set of decision rules from epoch k' onwards (e.g., $\big(X^{\pi}_{k'}, X^{\pi}_{k'+1}, \ldots, X^{\pi}_{K}\big)$), and its objective value is equivalent to Equation (11) but with the summation beginning at epoch k' and the expectation conditional on initial state $s_{k'}$.

3 Literature Review

The body of literature on electric vehicle routing is growing quickly. Our review first considers some of the seminal works in E-VRPs before concentrating specifically on those that consider public charging stations and dynamic solution methods. For a more in-depth review of the E-VRP literature, we refer the reader to Pelletier et al. (2016).

The Green VRP introduced by Erdoğan and Miller-Hooks (2012) is often cited as the origin of E-VRPs. The authors use mixed-integer-linear programming to assign routing and refueling decisions for a homogeneous fleet of alternative fuel vehicles. In the work, a number of simplifying assumptions are made that are difficult to justify for electric vehicles, such as that vehicles always fully restore their energy when they refuel and that refueling operations require constant time. The latter assumption was addressed in Schneider et al. (2014) who focus specifically on electric vehicle routing. They propose an E-VRP with time windows and capacity constraints (E-VRP-TW) for which they offer a heuristic solution. While still requiring full recharges, they relax the constant-time assumption for charging operations, instead assuming that the time required to perform these recharging operations is linear with the amount of energy to be restored. Desaulniers et al. (2016) offer exact solution methods for four variants of the E-VRP-TW and additionally relax the assumption on full recharging: two of the E-VRP-TW variants they address allow partial recharging operations. These operations are again assumed to take linear time with respect to the restored energy. In their work on the E-VRP with nonlinear charging functions, Montoya et al. (2017) demonstrate that the assumption of linear-time recharging operations can lead to infeasible or overly-expensive solutions. The aforementioned studies assume a homogeneous fleet of vehicles, but heterogeneous fleets consisting (at least in part) of EVs have also been considered in a number of studies, including Goeke and Schneider (2015); Hiermann et al. (2016); Hiermann et al. (2019); and Villegas et al. (2018). A number of additional E-VRP variants, such as those considering location-routing (Schiffer et al. 2018), congestion (Omidvar and Tavakkoli-Moghaddam 2012), and public transportation (Barco et al. 2017) have also been studied.

Despite the breadth of variants addressed, a common shortcoming in existing E-VRP studies is the lack of consideration of access to public charging infrastructure. Instead, studies generally make one of the two following assumptions: that the vehicles charge only at the depot (they adopt the private-only recharging strategy); or they allow extradepot (public-private) recharging, but the extradepot stations behave as if they were private, allowing EVs to begin charging immediately upon their arrival. Operating under the latter assumption promises solutions that are no worse than those under the former, as it simply enlarges the set of CSs at which EVs may charge. However, in reality, the adoption of the public-private recharging strategy introduces uncertainty and risk, and current E-VRP solution methods do not address this. As evidenced in Villegas et al. (2018), this leads companies to prefer the private-only approach despite results suggesting that the public-private approach offers better solutions. Having access to a dynamic routing solution capable of responding to uncertainty may encourage companies to utilize public CSs. However, such solutions are lacking, as research on dynamic routing of EVs is limited.

In a recent review of the dynamic routing literature by Psaraftis et al. (2016), the authors note the current dearth of dynamic EV routing research, citing only one study (Adler and Mirchandani 2014) and acknowledging that it would be more properly classified as a path-finding problem than a VRP. In that study, Adler and Mirchandani (2014) consider EVs randomly requesting routing guidance and access to a network of battery-swap stations (BSSs). The work addresses the problem from the perspective of the owner of the BSSs, aiming to minimize average total delay for all vehicles requesting guidance. Because reservations are made for EVs as they request and receive routing guidance, waiting times for the EVs at the BSSs are known in advance, eliminating uncertainty in their total travel time. A more recent study by Sweda et al. (2017) considers a path-finding problem in which a vehicle travels from an origin to a destination on a network with CSs at every node, and where each CS has a probability of being available and some expected waiting time (known a priori to the planner) if it is not. The decision maker dynamically decides the vehicle's path and recharging decisions so as to arrive at the destination as quickly as possible. The authors provide analytical results, including the optimal a priori routing policy. However, similar to Adler and Mirchandani (2014), the problem addressed more closely aligns with the family of path-finding problems rather than VRPs. Thus, a review of the literature reveals little existing work on dynamic E-VRPs. We seek to contribute to this domain with our research here.

4 Fixed Routes in the E-VRP-PP

We call a *fixed route* a complete set of routing and charging instructions from some origin node to a destination node, through some number of CSs and customer locations, that is prescribed to a vehicle prior to its departure. We often think of fixed routes in the context of static routing (e.g. Campbell and Thomas (2008)), but we can map them to dynamic routing as well, where a fixed route represents a predetermined sequence of actions from some state s_k to a terminal state s_K. The expected cost of a fixed route is the expected sum of the costs of these actions, which we can use as an estimate of the expected cost-to-go from s_k, the route's starting state. This makes fixed routes a useful tool in solving dynamic routing problems, such as the E-VRP-PP. In the coming sections, we show how fixed routes can be used to develop both static and dynamic policies, as well as establish dual bounds. In this section, we first formalize the concept of fixed routes for the E-VRP-PP in §4.1, then introduce a decomposition that facilitates the search for good fixed routes in §4.2. The decomposition is conducive to solving via classical methods from static and deterministic routing, which we detail in §4.3 and §4.4.

4.1 Definitions and AC Policies

Fixed Routes. In the E-VRP-PP, a fixed route consists of a set of instructions specifying the order in which to visit nodes $v \in V$ and to which $\tilde{q} \in \mathcal{Q}$ to charge when visiting CS nodes. Formally, we define a fixed route p to be a sequence of directions: $p = (p_1, p_2, \dots, p_{|p|})$, where each direction $p_j = (p_j^i, p_j^q)$ is a location-charge pair, similar to an action. Let us consider a vehicle in the state $s_1 = (t_{0,3}, 3, Q - e_{0,3}, \{1, 2\}, 1)$, as in Figure 3. We might consider the fixed route

$$p = ((3, Q - e_{0,3}), (2, Q - e_{0,3} - e_{3,2}), (4, Q - e_{0,3} - e_{3,2} - e_{2,4}), (4, \tilde{q}), (1, \tilde{q} - e_{4,1}), (0, \tilde{q} - e_{4,1} - e_{1,0})) \,, \tag{12}$$

which consists of routing instructions to the remaining unvisited customers $\bar{\mathcal{N}}_1$, as well as a detour to CS 4 at which the vehicle charges to some energy $\tilde{q} \in \mathcal{Q}$.

Fixed-Route Policies. The sequence of directions comprising a fixed route p constitutes a *fixed-*

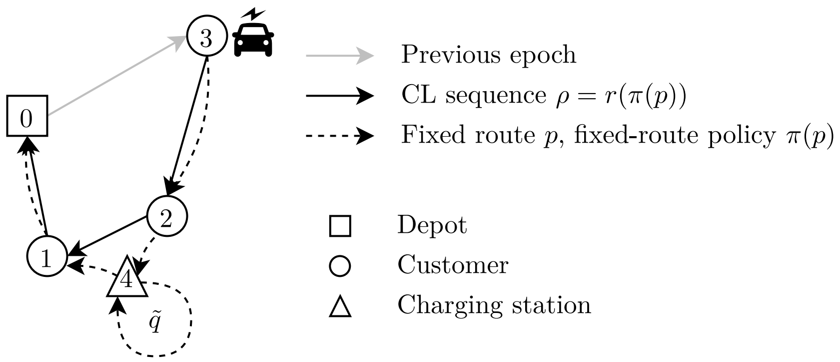

Figure 3: Shown is an EV that relocated from the depot to customer 3 in epoch 0. The CL sequence ρ (solid black arrows) considered by the vehicle from its current state is $(3, 2, 1, 0)$. The fixed route p (dashed black arrows) includes a detour to CS 4 where it charges to $\tilde{q}$, as indicated by the self-directed arc at 4 (p is given by equation (12)).

route policy, equivalently, a *static policy* $\pi(p) \in \Pi$, which is defined by decision rules

$$X_k^{\pi(p)}(s_k) = \begin{cases} p_{j^\star-1} & i_k \in \mathcal{C}' \wedge \left(p^q_{j^\star} > p^q_{j^\star-1} \wedge z_k > \psi_{i_k}\right) \\ p_{j^\star} & \text{otherwise,} \end{cases} \tag{13}$$

where $j^\star$ is the index of the next direction in p to be followed by the vehicle. Specifically, for state s_k, $j^\star$ is the index in p such that $\left(i_k = p^i_{j^\star-1} \wedge q_k = p^q_{j^\star-1} \wedge \bar{\mathcal{N}}_k = \left(\bigcup_{l=j^\star}^{|p|} p^i_l\right) \setminus \mathcal{C}'\right)$. Equation (13) simply directs the vehicle to follow the fixed route p. The first case addresses waiting actions which are not explicitly outlined in the routing instructions. If the vehicle encounters a queue at a CS at which it is instructed to charge, fixed-route policies dictate that it simply wait until a charger is available. The second case handles all other decision making as instructed by the fixed route. If we again consider the example in Figure 3 with fixed route p given by equation (12), then the corresponding fixed-route policy $\pi(p)$ would consist of the following sequence of decision rules and resulting actions:

$$\pi(p) = \Big(X_1^{\pi(p)}(s_1) = (2, q_1 - e_{3,2}), X_2^{\pi(p)}(s_2) = (4, q_2 - e_{2,4}), X_3^{\pi(p)}(s_3) = (4, \tilde{q})^*,$$
$$X_4^{\pi(p)}(s_4) = (1, \tilde{q} - e_{4,1}), X_5^{\pi(p)}(s_5) = (0, q_5 - e_{1,0})\Big).$$

The asterisk on action $(4, \tilde{q})$ in the third epoch indicates the potential presence of an additional prior epoch: if the vehicle arrives to CS 4 and there is a queue, then the vehicle must first wait before it can charge; in this case, an epoch $X_3^{\pi(p)}(s_3) = (4, q_2 - e_{2,4})$ is inserted, and the subsequent decision rules are shifted back (e.g., $X_5^{\pi(p)}$ becomes $X_6^{\pi(p)}$). Note that if we know waiting times in advance (see §6.2), then the existence of a waiting epoch would be known a priori.

From a state s_k, the set of all fixed-route or static policies is $\Pi^{\mathrm{S}} \subseteq \Pi$, defined as the set $\Pi^{\mathrm{S}} = \{\pi(p) \in \Pi \,|\, p \in P\}$ where P is the set of all feasible fixed routes (for a formal definition of P, see §A). We refer to such policies as static, because they offer no meaningful way in which to respond to uncertainty.

Paths and Compulsory-Location Sequences. Given a fixed-route policy $\pi(p) \in \Pi^{\mathrm{S}}$, let us denote by $R(\pi(p))$ the sequence of locations visited, which we call a *path*: $R(\pi(p)) = (p^i_j)_{j \in \{1,\ldots,|p|\}}$. In the above example, the path is $R(\pi(p)) = (3, 2, 4, 4, 1, 0)$. Notice that some of the locations in the path $R(\pi(p))$ must be visited by all valid fixed-route policies initialized from state s_1. Namely, all fixed-route policies have to include the vehicle's starting point ($p^i_1 = 3$), its ending point (the depot, $p^i_{|p|} = 0$), and the unvisited customers (1 and 2). We denote by $r(\pi(p))$ the subsequence of $R(\pi(p))$ consisting of only these locations: $r(\pi(p)) = (3, 2, 1, 0)$. In general,

$$r(\pi(p)) = p_1^i \frown (p_j^i : p_j^i \in \mathcal{N})_{1<j<|p|} \frown p_{|p|}^i \tag{14}$$

("$\frown$" is the concatenation operator). We call sequences like $r(\pi(p))$ *compulsory-location* (CL) *sequences*. Paths $R(\pi(p))$ contain additional elements for charging operations that occur while traversing a CL sequence. If it were energy-feasible for a vehicle to traverse a CL sequence $r(\pi(p))$ directly, then there would be no need to perform charging operations along the way, so $r(\pi(p))$ would be equivalent to $R(\pi(p))$ (under a good policy). That is, we can think of fixed routes and paths as being the energy-feasible analogs to CL sequences.

Our current definition of CL sequences (14) requires a fixed route p. However, we may also consider CL sequences from the perspective of a vehicle that does not currently have a prescribed fixed route. Recall that CL sequences begin with the vehicle's current location, end with the depot, and must contain in between the unvisited customers. We define the set of all CL sequences from a state s_k as those in the set $\mathcal{R}(s_k) = \{i_k \frown S_i \frown 0 : \; S_i \in \mathrm{Sym}(\bar{\mathcal{N}}_k)\}$, where $\mathrm{Sym}(\bar{\mathcal{N}}_k)$ is the set of all permutations of unvisited customers $\bar{\mathcal{N}}_k$. For our example with a vehicle occupying state s_1 in Figure 3, the set of CL sequences is $\mathcal{R}(s_1) = \{(3, 1, 2, 0), (3, 2, 1, 0)\}$. In general, we refer to CL sequences in $\mathcal{R}(s_k)$ by $\rho = (\rho_1, \ldots, \rho_{|\rho|})$, where $\rho_1 = i_k$ and $\rho_{|\rho|} = 0$.

AC Policies. Without loss of optimality, we can restrict our search of fixed-route polices (and fixed routes) to those that always charge when visiting a charging station. We refer to policies meeting this criterion as *AC policies* $\Pi^{\mathrm{AC}} \subseteq \Pi^{\mathrm{S}}$ (AC for "always charge"). The set of AC policies is defined as $\Pi^{\mathrm{AC}} = \Pi^{\mathrm{S}} \setminus \Pi^{\mathrm{B}}$, where $\Pi^{\mathrm{B}} = \left\{\pi(p) \in \Pi^{\mathrm{S}} \,\middle|\, p \in P \wedge \left(\exists j \in \{2, \ldots, |p| - 1\} : \; p_j^i \in \mathcal{C} \wedge p_{j-1}^i \neq p_j^i \wedge p_j^i \neq p_{j+1}^i\right)\right\}$ is the set of static policies that include CS visits at which no charging is performed. Note that all static policies belong to either Π^{B} or Π^{AC} (but not both). We justify the restriction of static policies to AC policies in the proof of Proposition 1. Going forward, all mention of static or fixed-route policies, unless explicitly stated otherwise, refers to those that are AC.

Proposition 1 (Good fixed-route policies are AC Policies)**.** *For all static, non-AC-policies $\pi \in \Pi^S \backslash \Pi^{AC}$, there exists an AC policy $\pi^{AC} \in \Pi^{AC}$ whose objective value is no worse: $\tau(\pi^{AC}) \leq \tau(\pi)$.*

Proof. Proof. See §B. □

Following from Proposition 1, we restrict our search of fixed routes and fixed-route policies to those that are AC.

4.2 Decomposition of the E-VRP-PP

Because fixed routes are central to our development of solution methods for the E-VRP-PP, we seek ways to establish good fixed routes. To do so, we leverage a decomposition of the problem into routing and charging decisions. Let us assume a vehicle occupies some state s_k. For a given CL sequence $\rho \in \mathcal{R}(s_k)$, we may search over the corresponding set of fixed-route policies, $\Pi_\rho \subseteq \Pi^{\mathrm{AC}}$, where $\Pi_\rho = \{\pi(p) \in \Pi^{\mathrm{AC}} : r(\pi(p)) = \rho\}$. Note that it is possible that for some ρ, the set Π_ρ will be empty. That is, there may exist CL sequences such that there does not exist an energy-feasible way in which to traverse the sequence. We offer the following decomposition:

Proposition 2. *For AC policies beginning in a state s_k, the E-VRP-PP can be decomposed into routing and charging decisions with objective*

$$\min_{\pi(p)\in\Pi^{AC}} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi(p)}(s_{k'}))\right] = \min_{\rho\in\mathcal{R}(s_k)} \left\{\min_{\pi\in\Pi_\rho} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi}(s_{k'}))\right]\right\}. \tag{15}$$

Proof. Proof. See §C. □

A solution to Equation (15) is an optimal fixed route – equivalently, an optimal fixed-route policy – whose cost provides an estimate of the cost-to-go from the route's starting state s_k. We exploit this in the construction of routing policies as well as in the establishment of dual bounds, where it aids in the computation of the value of an optimal policy with perfect information.

4.3 Solving the Decomposed E-VRP-PP

To solve Equation (15) we employ a Benders-like decomposition, taking the outer minimization over CL sequences as the master problem and the inner minimization over charging decisions as the sub-problem. Specifically, we use a *Benders-based branch-and-cut* algorithm in which at each integer node of the branch-and-bound tree of the master problem, the solution is sent to the subproblem for the generation of Benders cuts. We discuss the master problem in §4.3.1 and the subproblem and the generation of cuts in §4.3.2.

4.3.1 Master Problem: Routing.

The master problem corresponds to the outer minimization of Equation (15) in which we search over CL sequences. CL sequences are comprised of elements in the set $\mathcal{O}_k = i_k \cup \bar{\mathcal{N}}_k \cup \{0\}$. The master problem approximates the cost of traversing a CL sequence $\rho \in \mathcal{R}(s_k)$ by its *direct-travel cost* $T_D(\rho) = \sum_{j=1}^{|\rho|-1} t_{\rho_j,\rho_{j+1}}$. This approximation gives the cost of traversing ρ without charging.

To search CL sequences, we use a subtour-elimination formulation of the TSP (Dantzig et al. 1954) over the nodes in the subgraph of $\mathcal{G}$ with vertex set $\mathcal{O}_k$. This yields the following master problem:

$$\text{minimize} \quad \sum_{i\in\mathcal{O}_k}\sum_{j\in\mathcal{O}_k} t_{ij}x_{ij} + \theta \tag{16}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{O}_k} x_{ij} = 1, \quad \forall i \in \mathcal{O}_k \tag{17}$$

$$\sum_{i\in\mathcal{O}_k} x_{ij} = 1, \quad \forall j \in \mathcal{O}_k \tag{18}$$

$$x_{ii} = 0, \quad \forall i \in \mathcal{O}_k \tag{19}$$

$$\sum_{i,j\in S} x_{ij} \le |S| - 1, \quad \forall S \subset \mathcal{O}_k, |S| \ge 2 \tag{20}$$

$$x_{ij} \in \{0,1\}, \theta \ge 0 \tag{21}$$

If the vehicle is not initially at the depot (if $i_k \neq 0$), we add the constraint

$$x_{0i_k} = 1, \tag{22}$$

and set $t_{0i_k} = 0$, ensuring that the CL sequence ends at the depot and begins at i_k. Constraints (17) and (18), respectively, ensure that the vehicle departs from and arrives to each node exactly once; and constraints (19) prohibit self-directed arcs. Constraints (20) are the subtour elimination constraints, and (21) defines variables' scopes.

The binary variables x_{ij} take value 1 if node i immediately precedes node j in the CL sequence. A solution to the master problem is denoted by $\mathbf{x}$, and we call the subset of variables that take nonzero value $\mathbf{x}_\rho = \{x_{ij} | x_{ij} = 1\}$. The variables $\mathbf{x}_\rho$ define a CL sequence ρ, with $\rho_1 = i_k$ and all other ρ_i equal to the element in the singleton $\{j | x_{\rho_{i-1} j} \in \mathbf{x}_\rho\}$.

In addition to the direct travel cost of ρ, the objective function (16) contains the variable θ whose value reflects the additional cost of making the traversal of ρ energy feasible. To improve the master problem's estimation of this cost, we add valid inequalities for the minimum time that must be spent detouring to and recharging at charging stations:

$$\sum_{i \in \mathcal{O}_k} \sum_{j \in \mathcal{O}_k} (e_{ij} x_{ij} + \tilde{e}_{ij} y_{ij}) - q_k \leq e_R \tag{23}$$

$$\frac{1}{r^\star} e_R \leq t_R \tag{24}$$

$$\frac{1}{Q} e_R \leq n_e \tag{25}$$

$$\sum_{i \in \mathcal{O}_k} \sum_{j \in \mathcal{O}_k} y_{ij} \geq n_e \tag{26}$$

$$y_{ij} \leq x_{ij} \qquad \forall i, j \in \mathcal{O}_k \tag{27}$$

$$\sum_{i \in \mathcal{O}_k} \sum_{j \in \mathcal{O}_k} \tilde{t}_{ij} y_{ij} \leq t_D \tag{28}$$

$$\theta \geq t_D + t_R \tag{29}$$

$$y_{ij} \in [0,1]; e_R, n_e, t_D, t_R \geq 0 \tag{30}$$

We have introduced new variables: t_D and t_R are the minimum time spent detouring to and recharging at charging stations, e_R is the minimum energy that must be recharged, n_e is the minimum number of recharging events that must occur, and y_{ij} are variables indicating whether a recharging event should occur between nodes i and j. We have also introduced parameters $\tilde{e}_{ij}$ and $\tilde{t}_{ij}$ equal to the minimum energy and time to detour to a charging station between nodes i and j, as well as $r^\star$ which is the fastest rate at which the vehicle may replenish its energy across all charging stations (e.g., the slope of the first segment for the "Fast" charging function in Figure 1).

Equation (23) sets a lower bound for the amount of energy that must be recharged, and Equation (24) uses this to set a lower bound for the amount of time that the vehicle must spend recharging. Equation (25) sets a lower bound for the number of recharging events n_e that must occur, and Equation (26) requires the sum of insertion variables y_{ij} to be at least that amount. Equation (27) ensures that we only consider insertions along selected arcs. Equation (28) sets a lower bound for the time spent detouring to charging stations, and finally, Equation (29) uses the established bounds on the detouring and recharging times to increase the lower bound for θ. Note that in Equation (30), where we define scopes for the new variables, we have defined n_e and the y_{ij}s to be continuous. Although this is a less natural definition that results in a looser bound on θ, we find that this reduction in the number of integer (branching) variables leads to better performance.

4.3.2 Subproblem: Charging.

The master problem (16)-(30) produces a solution $\mathbf{x}_\rho$ that is passed to the subproblem. The subproblem is responsible for determining the optimal charging decisions along the sequence ρ and correcting, through the variable θ, the objective function value of the master problem associated with the solution $\mathbf{x}_\rho$. Call $Y^\star(\rho)$ the set of optimal charging decisions for sequence ρ. The decisions $Y^\star(\rho)$ include to which CSs to detour between which stops in the sequence and to what energy level to charge during these CS visits. Together, ρ and $Y^\star(\rho)$ constitute a fixed-route policy $\pi \in \Pi_\rho$. This problem of finding the optimal charging decisions given a CL sequence (the inner minimization of Equation (15)) is referred to as a *fixed-route vehicle charging problem*, or FRVCP (Montoya et al. 2017).

The subproblem will be one of two variants of the FRVCP, depending on the amount of information available to the decision maker. The amount of available information is known as the *information filtration* and is discussed in more detail in §6.2. If we assume the decision maker is operating under the natural filtration in which they can access all information that would naturally be available according to the problem definition in §2, then we solve the *FRVCP-N*. In the FRVCP-N, when we consider visiting a charging station $c \in \mathcal{C}$, in addition to the detouring and charging costs, we incur a cost equal to the *expected* waiting time at c. Alternatively, if we assume the decision maker is operating under the perfect information (PI) filtration, then we solve the *FRVCP-P*. With perfect information, the decision maker knows how long the vehicle must wait at every CS at every point in time. Hence, in the FRVCP-P, when we consider visiting a charging station c, we incur a cost equal to the *actual* waiting time as determined by realizations of queue dynamics.

Examples of the expected and actual waiting times considered by the decision maker in the FRVCP-N and FRVCP-P, respectively, are depicted in Figure 4. At left in the figure we show the case of the FRVCP-N. Expected waiting times are dependent on the most recent memory m of the queue: the lines correspond to different values for the number previously observed at the CS (m^l; indicated by line labels), with each line representing the expected waiting time at the CS as a function of the time since the memory was recorded (m^t). The memory-dependent waiting times follow from known dynamics and transient solutions for M/M/c queues. We rely on the software from Andersen (2020) for these calculations. Note that if the decision maker has no memory of the queue at a CS, then the estimate of the expected waiting time is constant and equal to the long-run expected waiting time of the queuing system, the value to which all of the transient expected waiting time curves converge (dashed line in left plot of Figure 4). At right is the case of the FRVCP-P. The piecewise function shows the actual time the vehicle would have to wait at a CS as a function of when it arrives. The discontinuities correspond to the times at which other customers arrive to the CS.

In general, we can model FRVCPs using dynamic programming. The formulation of this dynamic program (DP) for the subproblem is identical to the primary formulation for the E-VRP-PP outlined in §2, except we now operate under a more restricted action space $\mathcal{A}^{\mathrm{AC}}(s_k, \rho)$. This action space disallows non-AC policies, and it ensures that the vehicle follows the CL sequence ρ. We provide a formal definition of this restricted action space in the appendix, §D.

To solve the subproblem DP, we use the exact labeling algorithm for the FRVCP proposed by Froger et al. (2019). However, the FRVCP under consideration here requires discrete charging decisions and the inclusion of time-dependent waiting times at the CSs. We modify the labeling algorithm to account for these two additional features, which were not present in Froger et al.'s original formulation. The algorithm and our modifications to it are discussed in more detail in §4.4 and §E.

Optimality cuts. An optimal solution to an FRVCP is an optimal fixed route with CL sequence ρ. Call $T(\rho, Y^\star(\rho))$ the cost of the fixed route with CL sequence ρ and optimal charging decisions $Y^\star(\rho)$. If the direct-travel costs for ρ are $T_D(\rho)$, then the subproblem objective is $\theta_\rho := T(\rho, Y^\star(\rho)) - T_D(\rho)$, and

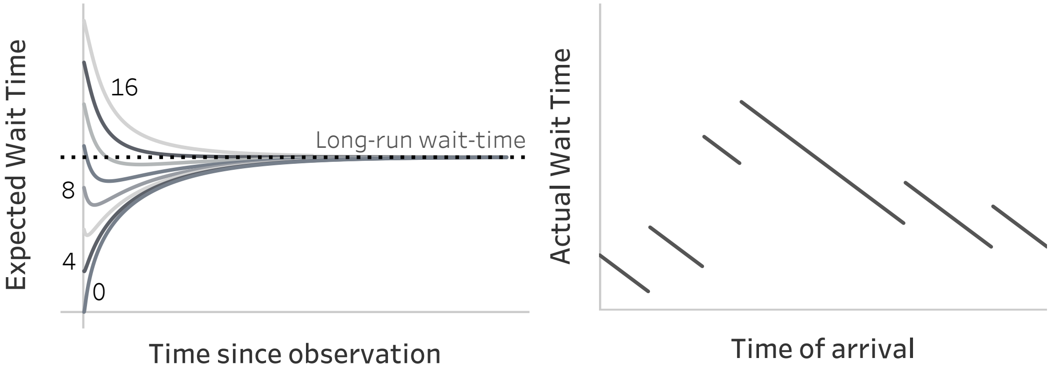

Figure 4: Waiting times at an extradepot CS under natural and perfect information filtrations. Under the natural filtration (left), the operator only has access to expected waiting times at the CS, which are dependent on the most recent memory m of the CS. The figure shows the expected waiting time as a function of the elapsed time since the memory was recorded (at m^t) for multiple values of the number of vehicles observed at the CS (m^l; indicated by line labels). Under the perfect information filtration (right), the operator is aware of the actual time they would have to wait at the CS.

we add to the master problem the following Benders optimality cuts:

$$\theta \geq \theta_\rho \left(\left(\sum_{x \in \mathbf{x}_\rho} x \right) - (|\mathbf{x}_\rho| - 1) \right) \tag{31}$$

The constraint works by ensuring that if the master problem selects sequence ρ by setting all $x \in \mathbf{x}_\rho$ to 1, then $\theta \geq \theta_\rho$. Otherwise, the right-hand side of (31) is at most 0, which is redundant given the non-negativity constraint on θ (21).

The optimality cuts in Equation (31) apply only to the complete CL sequence ρ. Cuts that apply to multiple sequences would be stronger, having the potential to eliminate more nodes from the branch-and-bound tree of the master problem. To build more general cuts, we consider *substrings* (consecutive subsequences) of ρ of length at least two. For example, for customer set $\mathcal{N} = (1, 2, 3)$ and CL sequence $\rho = (0, 2, 3, 1, 0)$, we would consider substrings $(0, 2)$, $(0, 2, 3)$, $(0, 2, 3, 1)$, $(2, 3)$, $(2, 3, 1)$, $(2, 3, 1, 0)$, $(3, 1)$, $(3, 1, 0)$, and $(1, 0)$. Denote the set of substrings of ρ by $\mathcal{P}_\rho$. We define the set $\bar{\mathcal{P}}_\rho \subseteq \mathcal{P}_\rho$ consisting of those substrings which cannot be traversed without charging: $\bar{\mathcal{P}}_\rho = \left\{ \sigma \in \mathcal{P}_\rho \middle| e^\star_{\sigma_1} < e_{\sigma_1 \sigma_2} + e_{\sigma_2 \sigma_3} + \cdots + e_{\sigma_{|\sigma|-1} \sigma_{|\sigma|}} \right\}$, where $e^\star_j = \max_{c \in \mathcal{C}} (Q - e_{cj})$ is the max charge an EV can have when departing location j. For each $\sigma \in \bar{\mathcal{P}}_\rho$, as we did for the complete sequence ρ, we compute $\theta_\sigma = T(\sigma, Y^\star(\sigma)) - T_D(\sigma)$, the difference between the minimum cost of an energy-feasible route through σ and its direct-travel costs. We then add cuts

$$\theta \geq \theta_\sigma \left(\left(\sum_{x \in \mathbf{x}_\sigma} x \right) - (|\mathbf{x}_\sigma| - 1) \right) \qquad \forall \sigma \in \bar{\mathcal{P}}_\rho,$$

where $\mathbf{x}_\sigma$ are the nonzero variables from the master problem solution $\mathbf{x}$ that define the substring σ.

To compute the values $T(\sigma, Y^\star(\sigma))$ for substrings $\sigma \in \bar{\mathcal{P}}_\rho$, we follow a process similar to the one used to compute $T(\rho, Y^\star(\rho))$ for the full sequence ρ. That is, $T(\sigma, Y^\star(\sigma))$ is the cost of the fixed route resulting from solving an FRVCP on the substring σ. However, we need to modify the FRVCP from the original model solved for ρ. First, of course, the CL sequence for which we solve for charging decisions is now σ instead of ρ. Next, for any substring $\sigma' \in \bar{\mathcal{P}}'_\rho = \{\sigma \in \bar{\mathcal{P}}_\rho | \sigma_1 \neq \rho_1\}$ that begins from a different location than ρ does, the time and charge at which the route begins are unknown. This is because prior to visiting σ'_1 along the sequence ρ, the EV may have stopped to charge. Having an unknown initial time means we can no longer solve an FRVCP with time-dependent waiting times, because when considering the insertion of a charging station into the route, we cannot say at what time the EV would arrive. In this case, in order to produce a conservative bound on the time required to travel the substring σ', we assume that all waiting times at charging stations are zero. Analogously, to account for unknown initial charge, we assume that we begin with the maximum possible charge ($e^\star_{\sigma'_1}$).

Feasibility cuts. If no feasible solution exists to the FRVCP for the CL sequence ρ, then it is impossible to traverse the sequence in an energy feasible manner, so the objective of the subproblem is infinite ($\theta_\rho = \infty$). This corresponds to the case where no fixed-route policy with CL sequence ρ exists ($\Pi_\rho = \emptyset$). In this case we add a feasibility cut eliminating the sequence ρ from the master problem: $\sum_{x \in \mathbf{x}_\rho} x \leq |\mathbf{x}_\rho| - 1$.

As we did for optimality cuts, we look to introduce stronger, more general feasibility cuts that may eliminate additional solutions in the master problem. We consider the substrings obtained by successively removing the first element in the sequence ρ. For each substring, we resolve an FRVCP, and if no feasible solution exists, add an optimality cut of the form $\sum_{x \in \mathbf{x}_{\rho'}} x \leq |\mathbf{x}_{\rho'}| - 1$, where ρ' is the substring $(\rho_j, \rho_{j+1}, \ldots, \rho_{|\rho|})$ formed by removing the first $j-1$ elements of ρ, and $\mathbf{x}_{\rho'}$ is the set of corresponding nonzero variables from the solution to the master problem. We continue this process until the sequence ρ' is reduced to length one or until we find a feasible solution for the FRVCP for ρ'. In the latter case we may stop, because a feasible solution will also exist for any substring of ρ'. As for the optimality cuts, we again assume that the initial charge when solving the FRVCP for a sequence ρ' is $e^\star_{\rho'_1}$. However, unlike for the optimality cuts, time-dependence is irrelevant, because we are simply searching for energy-feasibility of traversing ρ'. We may ignore waiting times completely and assume they are all zero.

4.4 Solving the FRVCP

The FRVCP entails the prescription of charging decisions for an electric vehicle following a fixed CL sequence such that traveling the sequence is energy feasible. The objective is to minimize the time required to reach the last node in the sequence. Froger et al. (2019) propose an exact algorithm to solve the FRVCP when the charging functions are concave and piecewise-linear and the charging decisions are continuous. In their implementation, waiting times at charging stations are not considered. We modify the algorithm to accommodate discrete charging decisions and time-dependent waiting times at the charging stations. These modifications are described in the appendix, §E. We provide here a brief overview of the algorithm.

To find the optimal charging decisions for a given CL sequence ρ, the FRVCP is reformulated as a resource-constrained shortest path problem. The algorithm then works by setting labels at nodes on a graph $\mathcal{G}'$ which reflects the vehicle's possible movements along ρ (see Figure 5). Labels are defined by *state-of-charge* (SoC) *functions*. (To maintain consistency with Froger et al. (2019), we continue to use the term "state-of-charge" here, which refers to the relative amount of charge remaining in a vehicle's battery, such as 25%; however, in general we measure the state of the battery in terms of its actual energy, such as 4 kWh.) SoC functions are piecewise-linear functions comprised of *supporting points* $z = (z^t, z^q)$ that describe a state of arrival to a node in terms of time z^t and battery level z^q. See Figure 6 for an

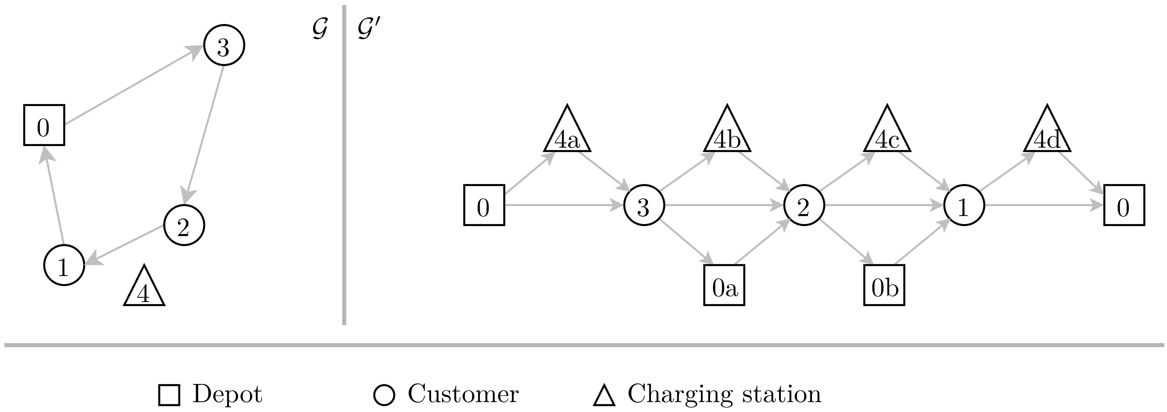

Figure 5: Left is an example of an original graph $\mathcal{G}$ for the E-VRP-PP. The gray path in the figure shows a CL sequence ρ. Right shows the corresponding modified graph $\mathcal{G}'$ used to model and solve the FRVCP, which includes a node for each possible CS visit.

example.

During the algorithm's execution, labels are extended along nodes in the graph $\mathcal{G}'$. Whenever a label is extended to a charging station node, we create new supporting points for each possible charging decision. Consider Figure 6, which depicts this process when extending a label along the edge from node 0 to node $4a$ in Figure 5. Initially there is only one supporting point, corresponding to the EV's arrival to CS 4 directly from the depot. That supporting point $z_1 = (t_{0,4}, Q - e_{0,4})$ is depicted by the black diamond in the left graph of Figure 6. We then consider the set of possible charging decisions at that CS. The right graph of Figure 6 shows the charging function at CS 4 with circles for the set of charging decisions $\mathcal{Q}$ for this example. Only the black circles q'_1 and q'_2 are valid charging decisions, however, since the others are less than z_1^q the vehicle's charge upon arrival to CS 4. For each valid charging decision, we add a supporting point to the SoC function (left), whose time and charge reflect the decision to engage in the charging operation. The figure shows this explicitly for the new supporting point z_3 corresponding to charging decision q'_2.

We continue to extend labels along nodes in $\mathcal{G}'$ until the destination node $\rho_{|\rho|} = 0$ is reached, whereat the algorithm returns the earliest arrival time of the label's SoC function. Bounds on energy and time are established in pre-processing and are used alongside dominance rules during the algorithm's execution in order to improve its efficiency. For complete details on the algorithm, we refer the reader to Froger et al. (2019).

With our modifications (§E), we can use the labeling algorithm to solve FRVCPs and create energy-feasible fixed routes for the E-VRP-PP. In the coming sections, we demonstrate the application of fixed routes in the construction of static and dynamic policies and in the establishment of dual bounds.

As we demonstrate in this work, the labeling algorithm from Froger et al. (2019) serves as a strong foundation upon which other researchers may build in order to solve their own variants of electric vehicle recharging problems. However, the implementation of this algorithm is non-trivial and may stand as a barrier for researchers interested in E-VRPs. In an attempt to remove this barrier, we offer a free and open-source implementation of the algorithm in a Python package called *frvcpy* (Kullman et al. 2020).

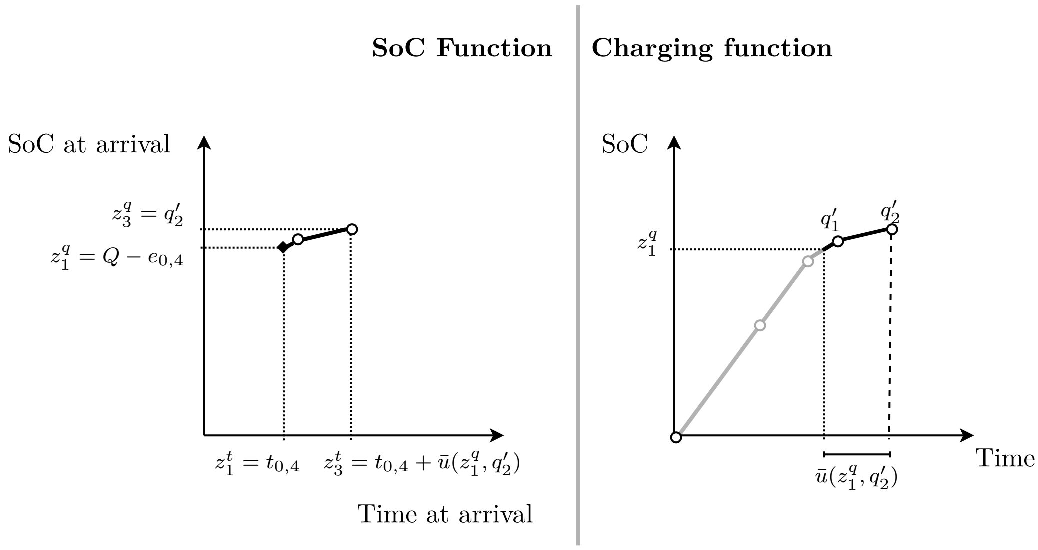

Figure 6: Depicting the creation of new supporting points at CS nodes for the case of node 4a in Figure 5. Left shows the SoC function at node 4a. The initial supporting point is the black diamond ($z_1 = (t_{0,4}, Q - e_{0,4})$). We create additional supporting points (z_2 and z_3, circles) for each possible charging decision. Possible charging decisions q_1' and q_2' are the black circles in the charging function (right graph). Axis labels on the SoC function for the new supporting point z_3 show how it is created from the charging decision q_2'.

5 Policies

In this section, we describe routing policies to solve the E-VRP-PP. We divide our discussion into static policies and dynamic policies. These classes of policies differ in when they make decisions and their use of exogenous information. We begin by describing static policies, whose decisions are made in advance and do not change in response to exogenous information. We then describe dynamic policies, which may use exogenous information to inform their decision making at each epoch. A tabular summary of the policies is provided in Table 1.

5.1 Static Policies

The decomposition in Proposition 2 provides a convenient way to find the optimal static policy. This is the first policy we propose to solve the E-VRP-PP. Then, because solving for an optimal static policy is computationally expensive, we also consider an approximation which we call the TSP Static policy. For both, following from Proposition 1, we restrict our search to only those static policies that are AC.

5.1.1 Optimal Static Policy.

An optimal static policy represents the best performance a decision maker can achieve when unable to respond dynamically to uncertainty. This serves as an upper bound on the optimal policy, since $\Pi^{AC} \subseteq \Pi$. For the E-VRP-PP, we can find such a policy by solving the nested minimization of equation (15); this solution produces an optimal fixed route from which an optimal fixed-route policy can be constructed. To solve equation (15), we use the Benders-based branch-and-cut algorithm described in §4.3.

5.1.2 TSP Static Policy.

Because solving Equation (15) to get an optimal static policy is computationally expensive, we introduce an approximation of the optimal static policy, the *TSP Static policy* π^{TSP}, that is easier to compute. The procedure to construct π^{TSP} is motivated by the decomposition in §4.2; however, we abbreviate our search over CL sequences, performing only a single iteration of the master and subproblems. The solution to a single iteration of the master problem is a CL sequence ρ^{TSP} representing the shortest Hamiltonian path over the unvisited customers and the depot. (We refer to this policy as the TSP Static policy, because when solving from the depot in the initial state, the shortest Hamiltonian path corresponds to the optimal TSP tour over $\mathcal{N} \cup \{0\}$.) We then optimally solve the FRVCP-N for ρ^{TSP} to generate an energy-feasible fixed route whose corresponding fixed-route policy we denote π^{TSP}.

5.2 Dynamic Policies

By definition, static policies do not use exogenous information to inform their decision making. The vehicle's instructions are prescribed in advance, and it simply follows them. Assuming exogenous information has value, these policies will be suboptimal. In this vein, we develop two dynamic policies leveraging rollout algorithms. As a benchmark, we also offer a myopic policy.

Rollout algorithms are lookahead techniques used in approximate dynamic programming to guide action selection. They may be classified by the extent of their lookahead, i.e., how far into the future they anticipate. Commonly implemented rollouts include one-step, post-decision (half-step), and pre-decision (zero-step). An m-step rollout requires the enumeration of the set of reachable states m steps into the future, constructing and evaluating a *base policy* at each future state to provide an estimate of the cost-to-go. This results in a trade-off: in general, deeper lookaheads and better base policies offer better estimations of the cost-to-go, but they require additional computation. Thus, as we consider deeper lookaheads, we are forced to consider simpler base policies. Here, we implement a pre-decision rollout with an Optimal Static base policy and a post-decision rollout with a TSP Static base policy.

5.2.1 Pre-decision Rollout of the Optimal Static Policy.

A pre-decision (or zero-step) rollout implements a base policy $\pi(s_k)$ from the pre-decision state s_k to select an action. The decision rule for pre-decision rollouts is simply to perform the action dictated by the base policy: $a^\star = X_k^{\pi(s_k)}(s_k)$. This strategy is also referred to as reoptimization, because the base policy is often determined by the solution to a math program that is repeatedly solved at each decision epoch. Following suit, we use the optimal static policy as our base policy, in each epoch following the procedure in §4.3 to determine the optimal fixed route from pre-decision state s_k and executing the first action prescribed by the fixed route. We call our pre-decision rollout of the optimal static policy *PreOpt*.

5.2.2 Post-decision Rollout of the TSP Static Policy.

Post-decision rollouts evaluate expected costs-to-go from post-decision states half of an epoch into the future. This is more computationally intensive than the procedure for pre-decision rollouts, because it requires the construction of a base policy from each post-decision state – of which there are $|\mathcal{A}(s_k)|$ – instead of only once from the pre-decision state s_k. Consider, for instance, action selection from some state s_k in which the vehicle just served a customer $i_k \in \mathcal{N}$. With $\bar{N} = |\bar{\mathcal{N}_k}|$ unvisited customers and C charging stations, there are up to $\bar{N} + C$ possible actions, corresponding to the relocation of the vehicle to each of these nodes. Finding the optimal static policy from each such post-decision state in each epoch

Table 1: Policy summaries. Static policies, to establish a fixed route to follow, perform computation only in the first epoch; dynamic policies do so at every epoch. (The Perfect Info policy is discussed in §6.)

Policy	Filtration	Type	Executed	Method
Perfect Info	$\mathbb{I}$	Static	First epoch only	$\rho^{\star}$ + FRVCP-P
Optimal Static	$\mathbb{F}$	Static	First epoch only	$\rho^{\star}$ + FRVCP-N
TSP Static	$\mathbb{F}$	Static	First epoch only	ρ^{TSP} + FRVCP-N
PreOpt	$\mathbb{F}$	Dynamic	Every epoch	$\rho^{\star}$ + FRVCP-N
PostTSP	$\mathbb{F}$	Dynamic	Each action, every epoch	ρ^{TSP} + FRVCP-N
Myopic	$\mathbb{F}$	Dynamic	Every epoch	$\min_{a\in\mathcal{A}(s_k)} C(s_k, a)$

is intractable. For this reason, we use the TSP static policy π^{TSP} as the base policy in our post-decision rollout. We call the post-decision rollout with the TSP Static base policy *PostTSP*.

Let $\mathcal{S}_{\mathrm{post}}(s_k) = \{s_k^a | a \in \mathcal{A}(s_k)\}$ be the set of reachable post-decision states. From each $s_k^a \in \mathcal{S}_{\mathrm{post}}(s_k)$, we solve for the shortest Hamiltonian path over the set $i_k^a \cup \bar{\mathcal{N}}_k^a \cup \{0\}$ to produce a CL sequence ρ^a, then solve the FRVCP-N for ρ^a to produce a fixed-route policy $\pi^{\mathrm{TSP}}(s_k^a)$ that serves as the base policy $\pi_{\mathrm{b}} = \pi^{\mathrm{TSP}}(s_k^a)$. The expected cost of this policy is the expected cost of the fixed route, given by $T(\rho^a, Y^{\star}(\rho^a))$. The post-decision rollout decision rule is then to select an action $a^{\star}$ solving

$$\min_{a\in\mathcal{A}(s_k)} \left\{ C(s_k, a) + \mathbb{E}\left[\sum_{i=k+1}^{K} C(s_i, X_i^{\pi_{\mathrm{b}}}(s_i)) \middle| s_k \right] \right\} = \min_{a\in\mathcal{A}(s_k)} \left\{ C(s_k, a) + T(\rho^a, Y^{\star}(\rho^a)) \right\}. \quad (32)$$

5.2.3 Myopic Policy.

As a benchmark for our other static and dynamic routing policies, we implement a myopic policy. Myopic policies ignore future costs in action selection, simply preferring actions with the cheapest immediate cost. More formally, myopic policies choose an action minimizing $\min_{a\in\mathcal{A}(s_k)} C(s_k, a)$. In practice, a myopic policy following this decision rule will result in exceptionally poor performance. For this reason, we bolster our myopic policy with the following rules: if all customers have been visited and the vehicle can reach the depot, we disallow all other actions; if the vehicle is at an available charging station and can charge, we require it to charge to full battery capacity; if the vehicle has arrived to a charging station where the current queue length is less than the expected queue length, the vehicle must queue; we disallow relocations to nodes other than customers, provided a customer can be reached; and the vehicle may not visit more than two extradepot charging stations between nodes in $\mathcal{N}_k \cup \{0\}$.

6 Dual Bounds

While we seek to produce policies that perform favorably relative to industry methods, gauging policy quality is hampered by the lack of a strong bound on the value of an optimal policy, a *dual bound*. Without an absolute performance benchmark, it is difficult to know if a policy's performance is "good enough" for practice or if additional research is required to improve the routing scheme. In §6.1 we first discuss a technology-based dual bound where we assume that the vehicle is powered by an internal-combustion engine. This bound ignores the need to detour, wait, and recharge at CSs. Assuming that these actions have non-negligible cost, this bound will likely be loose. In §6.2 we describe our efforts to establish a tighter dual bound using the expected value of an optimal policy with perfect information, i.e., the performance achieved via a clairvoyant decision maker.

With the aim of further tightening the dual bound, we develop nonlinear information penalties that punish the decision maker for using information about the future to which they would not naturally have access. These penalties are constructed using the fixed-route machinery from §4. We apply the penalties on action selection in a modified version of the decomposed problem (Equation (15)). To the best of our knowledge, our successful implementation of these penalties marks a first in the field of vehicle routing. However, this success is limited, because we could only apply the penalties to small instances; the computational costs to apply them on larger instances are prohibitive. As a result, the penalties did not provide practical value in tightening the dual bound on our real problem instances described in §7. To limit the length of this text, we present the detailed discussion of our information penalties in the appendix. See §F.

6.1 Conventional Vehicle Bound

To compute the optimal value with a conventional vehicle, we assume that the vehicle has infinite energy autonomy and no longer needs to recharge in order to visit all customers. We refer to this bound as the CV bound. The CV bound is a valid dual bound because it is a relaxation of the action space. Specifically, we relax the condition $(\exists c \in \mathcal{C} : a^q \geq e_{a^i c})$ in equation (4) and the condition $(q_k \geq e_{i_k a^i})$ in equation (5). These conditions are responsible for ensuring that the vehicle has sufficient charge to relocate. By relaxing these conditions, the vehicle can always relocate to an unvisited customer or a CS. Under a relaxation, the set of feasible policies increases: $\Pi \subseteq \Pi_{\mathrm{CV}}$, where Π is the set of feasible policies under the original action space and Π_{CV} is the set of feasible policies under the relaxed action space. Additionally, we know that there is an optimal policy $\pi^\star \in \Pi_{\mathrm{CV}}$ that does not visit any charging stations; see Proposition 3. The CV bound is the value of this policy.

Because the optimization can be restricted to policies that do not visit CSs, uncertainty in CS queues can be ignored. Consequently, we can further restrict the search to static policies and proceed as in §4.3. Without the need to perform charging operations, the subproblem objective (inner minimization of equation (15)) over charging decisions is zero, so an optimal solution is simply a CL sequence that minimizes direct-travel costs (the outer minimization). The resulting problem of finding this CL sequence is simply a classical traveling salesman problem (TSP) over the set of customers and the depot.

Proposition 3. *Let $\mathcal{A}_{CV}(s_k)$ be a relaxation of action space $\mathcal{A}(s_k)$ defined by the removal of conditions $(\exists c \in \mathcal{C} : a^q \geq e_{a^i c})$ in equation (4) and $(q_k \geq e_{i_k a^i})$ in equation (5). Further, let Π_{CV} be the set of feasible policies under $\mathcal{A}_{CV}$. Then there exists an optimal policy $\pi^\star \in \Pi_{CV}$ that does not visit any charging stations.*

Proof. Proof. See §G. □

6.2 Perfect Information Relaxation

Let $\mathscr{F}$ be the σ-algebra defining the set of all realizations of uncertainty. As in Brown et al. (2010), we define a *filtration* $\mathbb{F} = (\mathscr{F}_0, \ldots, \mathscr{F}_K)$ where each $\mathscr{F}_k \subseteq \mathscr{F}$ is a σ-algebra describing the information known to the decision maker from pre-decision state s_k. Intuitively, a filtration defines the information available to make decisions.

We will denote by $\mathbb{F}$ the *natural filtration*, i.e., the information that is naturally available to a decision maker. We describe any policy operating under the natural filtration as being *non-anticipative*. Given another filtration $\mathbb{G} = (\mathscr{G}_0, \ldots, \mathscr{G}_K)$, we say it is a *relaxation* of $\mathbb{F}$ if for each epoch k, $\mathscr{F}_k \subseteq \mathscr{G}_k$, meaning that in each epoch the decision maker has access to no less information under $\mathbb{G}$ than they do under $\mathbb{F}$. If $\mathbb{G}$ is a relaxation of $\mathbb{F}$, we will write $\mathbb{F} \subseteq \mathbb{G}$. In the current problem, for example, we could define a relaxation $\mathbb{G}$ wherein from a state s_k, the decision maker knows the current queue length at each CS.

In Brown et al. (2010), the authors prove that the value of the optimal policy under a relaxation of the natural filtration provides a dual bound on the value of the optimal non-anticipative policy. We use this result to formulate a bound on the optimal policy using what is known as the *perfect information* (PI) *relaxation.*

The perfect information relaxation is defined by the relaxation $\mathbb{I} = (\mathscr{I}_0, \ldots, \mathscr{I}_K)$ where each $\mathscr{I}_k = \mathscr{F}$. That is, the decision maker is always aware of the exogenous information that would be observed from any state; they are effectively clairvoyant, and there is no uncertainty. With all uncertainty removed, we can rewrite the objective function as

$$\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) \middle| s_0\right] = \mathbb{E}\left[\min_{\pi \in \Pi} \sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) \middle| s_0\right]. \tag{33}$$

Notice that the *perfect information problem* (33) can be solved with the aid of simulation. We may rely on the law of large numbers – drawing random realizations of uncertainty, solving the inner minimization for each, and computing a sample average – to achieve an unbiased and consistent estimate of the true objective value. Per Brown et al. (2010), this value serves as a dual bound on the optimal non-anticipative policy, a bound we refer to as the *perfect information bound*.

In the context of the E-VRP-PP, a clairvoyant decision maker would know in advance the queue dynamics at each extradepot CS at all points in time. This information is summarized in the right plot in Figure 4, which shows the time an EV must wait before entering service at an extradepot CS as a function of its arrival time. Then a realization of uncertainty, which we will call ω, contains the information describing such queue dynamics at all extradepot CSs across the operating horizon. Let us call the set of all possible realizations of queue dynamics Ω. Then to estimate the objective value of (33), we sample queue dynamics ω from Ω, grant the decision maker access to this information, solve for the optimal policy for each ω, and compute the sample average.

In the absence of uncertainty that results from having access to the information ω, the inner minimization can be solved deterministically. That is, all information is known upfront, so no information is revealed to the decision maker during the execution of a policy. As a result, there is no advantage in making decisions dynamically (epoch by epoch) rather than statically (making all decisions at time 0). This permits the use of static policies to solve the PI problem. Following from Proposition 1, which applies to static policies regardless of information filtration, we may restrict our search to AC policies. Further, as demonstrated in Proposition 2, we can decompose the search over AC policies into routing and charging decisions. As a result, we can rewrite the objective of the PI problem as

$$\mathbb{E}\left[\min_{\pi(p) \in \Pi^{\mathrm{AC}}} \sum_{k=0}^{K} C(s_k, X_k^{\pi(p)}(s_k)) \middle| s_0\right] = \mathbb{E}\left[\min_{\rho \in \mathcal{R}(s_0)} \left\{\min_{\pi \in \Pi_\rho} \sum_{k=0}^{K} C\left(s_k, X_k^\pi(s_k)\right)\right\} \middle| s_0\right]. \tag{34}$$

To solve the nested minimization for a given ω, we use the same decomposition and Benders-based branch-and-cut algorithm described in §4.2 and §4.3, respectively. Because we are operating under the perfect information filtration, the subproblem now corresponds to the FRVCP-P.

6.2.1 Tractability of the PI Problem

The value of the optimal policy with perfect information, *the PI bound*, is often computationally intractable to obtain. This is because the estimation of the expected value with perfect information entails repeated solutions to the inner minimization of equation (33), a challenging problem despite the absence of uncertainty. Even if the Benders-based method of §4.3.1 and 4.3.2 does not return an optimal solution for a given realization of uncertainty, we may still get a valid bound by using the best (lower) bound produced by the solver (Gurobi v8.1.1). The solver's bound, typically attained via linear relaxations to the master problem, serves as a bound on the value of an optimal policy with PI, and therefore also as a bound on an optimal policy. This mixed bound, effectively combining both an information and a linear relaxation, is weaker than a bound based on the information relaxation alone. In the computational experiments described in §7, when the bound results only from the information relaxation (i.e., we were able to solve all realizations of uncertainty to optimality), we denote the bound by an asterisk (*); otherwise, the bound is mixed.

7 Computational Experiments

To evaluate the performance of our routing policies, we assemble a testbed comprised of 102 real world instances. These instances are derived from the study by Villegas et al. (2018), in which French electricity giant ENEDIS rejected the public-private recharging strategy, citing concerns about uncertainty and risk at public CSs. We describe the generation of these instances in §7.1, then explore the results of our computational experiments in §7.2 with special emphasis on the comparison of private-only and public-private recharging strategies in §7.3 and an analysis of policy performance under potential future technologies in §7.4.

7.1 Instance Generation

In the study by Villegas et al. (2018), the authors explain that ENEDIS divides its maintenance and service operations into geographical zones. On the days of operation considered in their study, these zones contained between 54 and 167 customers each. For each zone, there is a set of technicians that serves the associated customers. The authors were responsible for assigning customers to and providing routing instructions for the technicians, subject to a number of constraints. In total across all zones, the solution by Villegas et al. included customer assignments for 81 technicians. It is from these 81 assignments that we create our instances. Specifically, our instances are derived from the subset of 34 of these 81 assignments whose shortest Hamiltonian cycle (TSP) cannot be traveled in a single charge by the EV proposed in their study. We assume worst-case energy consumption rates in order to provide the largest possible set of instances.

The charging stations included in our instances were taken from a database provided by the French national government (Etalab 2014). The database provides information on the number of chargers available at each charging station, as well as their maximum power output. We divide the charging stations into two types – moderate (power output less than 20 kW) and fast (greater than 20 kW) – that roughly correspond to the common Level 2 and Level 3 charging types. We assume the depot locations in the ENEDIS instances also contain fast charging terminals. Based on data from Morrissey et al. (2016), we set the mean service time μ_c of a CS c to be 26.62 minutes for fast CSs and 128.78 minutes for moderate CSs. The probability of departure from an occupied charger at the CS in a given minute is then $p_{c,\text{depart}} = 1/\mu_c$. For each of the 34 assignments, we consider a low, moderate, and high demand scenario, corresponding, respectively, to an average utilization u of 40%, 65%, and 90%. As an example, this means that under the high demand scenario the probability of all chargers being occupied when a vehicle arrives is 90%. Given a utilization u, the number of chargers at a CS ψ_c, and the probability of departure $p_{c,\text{depart}}$, we can compute the arrival probability according to $p_{c,\text{arrive}} = u \cdot \psi_c \cdot p_{c,\text{depart}}$. We assume the CSs have an infinite buffer so that a vehicle will never be stranded – it can always choose to wait. The charging functions for our CSs are those given in Montoya et al. (2017), which are piecewise linear and have breakpoints (changes in charge rate) at 85% and 95% of the vehicle's maximum battery capacity, which is $Q = 16$ kWh. We assume that the vehicle travels at a speed of 40 km/hr and consumes energy at a rate of 0.25 kWh/km. The set of energy levels to which the vehicle can charge $\mathcal{Q}$ consists of the charge function breakpoints as well as increments of 10%.

With 34 technician assignments and three demand scenarios for each, we have a primary testbed of 102 instances (assignment-demand pairs). These instances have between 8 and 26 customers (with an average of 16) and 6 and 79 extradepot charging stations (with an average of 49). The instances are publicly available at VRP-REP (Mendoza et al. 2014) under VRP-REP-ID: 2019-0004[1]. Results over this set of instances are described in §7.2. To compare with the industry-standard private-only recharging strategy, we also consider a "private-only" scenario of each technician assignment in which we remove all extradepot CSs. These are not included in the set on VRP-REP, since they can be easily reproduced from the primary instances. Discussion of this comparison to the private-only recharging strategy is in §7.2.

For each of the 102 primary instances, we seek to establish the PI bound and the expected objective value for each of our policies. To do so, we take the sample average of their objective values over 50 realizations of uncertainty. To generate a realization of uncertainty ω for an instance, we use a queue simulator to generate a day's worth of queue events for each extradepot CS c according to its queue parameters ($p_{c,\text{arrive}}$, $p_{c,\text{depart}}$, ψ_c). Then if the vehicle visits c at some time during the policy's execution, it observes the state of the queue according to ω.

7.2 Results on Primary Instances

We divide the discussion of the results on our primary instances as follows. First, we compare the two dual bounds proposed in §6. We then investigate policies' performances, first giving a brief overview, then comparing static and dynamic policies, and lastly comparing the policies to the PI bound. Finally, we comment on the computational effort to perform these experiments.

Comparing dual bounds.

We begin our analysis by comparing the optimal value of performing service with a CV, the *CV bound*, with the PI bound. To establish the PI bound for each instance we take the average over 50 realizations of uncertainty. Across the 5100 computational experiments for the PI bound (102 instances with 50 realizations of uncertainty for each), we are able to solve the PI problem to optimality in 3582 of them: 1637 in low, 1172 in moderate, and 673 in high. For the CV bound, since there is no uncertainty, we need solve it only once for each technician assignment. We were able to solve for the optimal CV bound for 34/34 assignments, yielding a CV bound for each instance.

Figure 7 offers a comparison of the PI bound to the CV bound. Instances for which we were able to optimally solve the PI problem for all 50 realizations of uncertainty are marked with an asterisk. We

[1]The instances currently have private visibility. We will update visibility to public after completion of the review process.

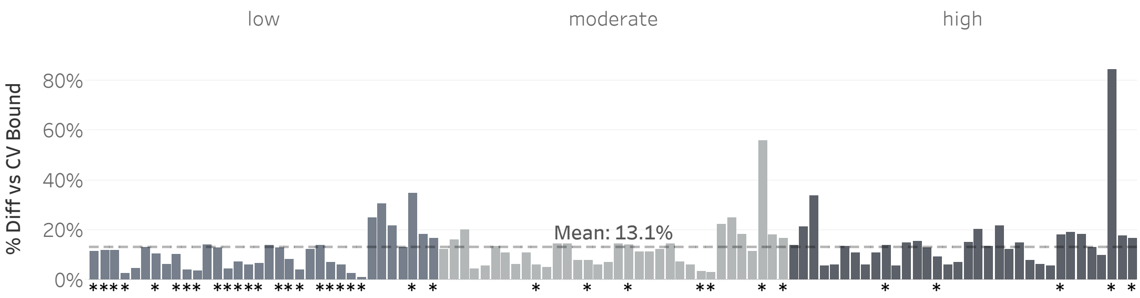

Figure 7: Comparing dual bounds. The figure contains a bar for each instance showing the relative performance difference between the CV and PI bounds. Instances for which the PI bound was solved to optimality for all 50 realizations of uncertainty are marked with an asterisk.

find that the PI bound is a significantly better dual bound than the CV bound, offering an improvement of 13.1% on average. Going forward, the reported performance gaps for our policies are stated relative to the PI bound. We note that the PI bound will offer a better gauge of policy performance for the lower demand instances, since in these cases the bound more frequently represents purely an information relaxation. In contrast, because the PI problems are more difficult to solve to optimality for the high demand instances, the bound increasingly also incorporates a linear relaxation. Thus the PI bound for these instances may be looser.

Having a tighter dual bound allows us to make stronger statements about the goodness of our routing policies, and, in general, suggests that there is value in the effort to establish the PI bound. More broadly, the gap between the CV and PI dual bounds also lends support to the notion that E-VRPs should indeed be considered a distinct family of problems from conventional VRPs.

Summary of Policy Performance.

To assess policies' performance on an instance, we average over 50 realizations of uncertainty, as we did to establish the PI bound. In computing the optimal static policy, we are able to solve equation (15) exactly in 40/102 instances. For the remainder, we use the best known solution (BKS) after three hours. To execute the PreOpt policy, in the first epoch we use the solution found by the optimal static policy, then allow two minutes to resolve the optimal static policy at all subsequent epochs, taking the best solution after two minutes if the optimal solution is not found in that time. Note that the optimal static policy need only be recomputed in epochs following the observation of (non-deterministic) exogenous information. For example, if in the first epoch the optimal static policy dictates relocating from the depot to a customer, then in the subsequent epoch the vehicle can continue to follow the optimal static policy without recomputing it, as no additional information was observed when it arrived to the customer. In the tables and figures that follow, unless noted otherwise, units are minutes.

As seen in Figure 8, we find that our route-based policies are competitive with one another, while the myopic policy serves as a distant upper bound. This contrast between the performance of our route-based policies and the myopic policy demonstrates the value in route-planning and the anticipation of charging station queues. Further, we find that the route-based policies are competitive with the PI dual bound, especially in the low and moderate demand scenarios (a more detailed discussion of policies' performance relative to the PI bound is below). As expected from queuing theory, the objective values of our policies increase with the demand for extradepot CSs. Of our policies, PreOpt performs the

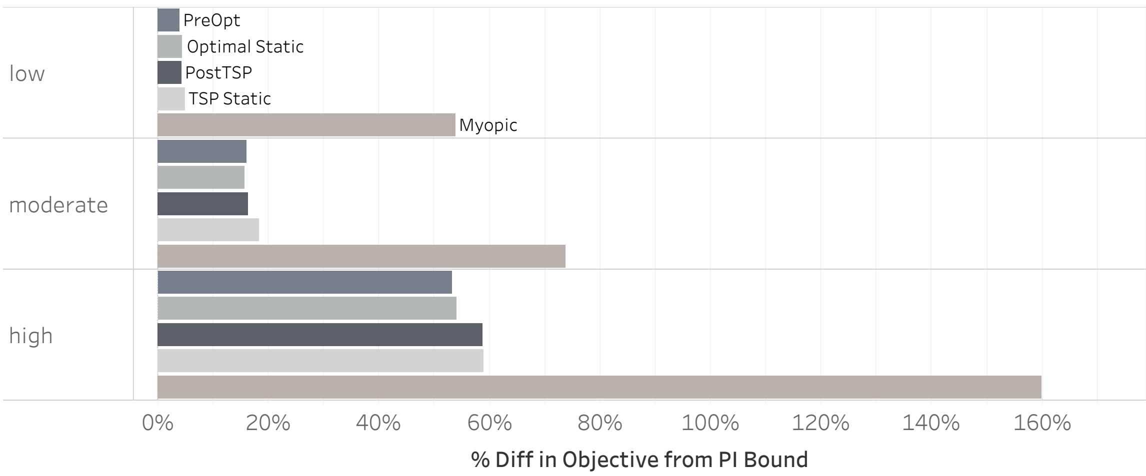

Figure 8: Visual summary of policy performance relative to the PI bound. Each bar is an average over the 34 technician assignments, with 50 realizations of uncertainty for each.

best on average, followed by the optimal static policy, PostTSP, then TSP Static. Comparing static policies, across demand scenarios, the optimal static policy offers on average a 3% improvement over the TSP static policy. The difference between our dynamic policies is similar, with PreOpt offering a 2% improvement over PostTSP across demand scenarios.

Performance of static vs. dynamic policies.

Figure 9 depicts the advantage that dynamic policies stand to offer over static policies – namely, that their additional flexibility in making routing decisions should yield improvements in objective values. We find this to be true here, with dynamic policies exhibiting a small edge over static policies, outperforming them by 0.5% on average. We see that this edge is largely attained through a reduction in waiting times, which outweighs an observed increase in travel times. These observations align with intuition. One would expect that static policies, which must wait at extradepot CSs regardless of observed queue length, would wait longer on average than dynamic policies, which can choose to balk CSs if queues are long. Consequently, relative to static policies, which wait in queue, dynamic policies should spend more time traveling as they explore additional CSs.

As shown in Figure 9, the objective performance of dynamic and static policies is surprisingly similar. We note that the decision maker only realizes new actionable information in a small subset of epochs – namely, those in which they arrive to an extradepot CS. If more information was revealed to the decision-maker during policy execution, it is likely that the performance of static and dynamic policies would diverge. We explore this idea in §7.4.

Policy performance relative to PI bound.

Table 2 compares policies' performance to the value of the optimal policy with perfect information. We find that on average our best policy is within 10% in the low and moderate demand scenarios, and within 24.4% overall. As seen in Figure 8, the gap between our routing policies and the PI bound widens as demand for extradepot CSs increases, from an average of 4.4% under the low demand scenario to 56.2% under the high demand scenario. This increasing gap to the PI bound is due in part to the fact that

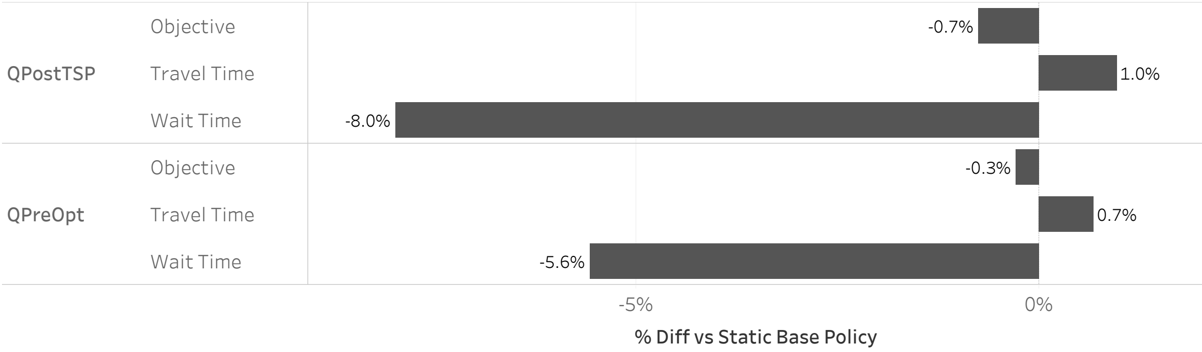

Figure 9: Comparing dynamic policies to the static base policies from which they are built. Top panel shows the percent difference in various metrics of PostTSP from the TSP static policy; bottom shows the same for PreOpt relative to the optimal static policy. Values reflect averages over all instances.

under higher demand scenarios, the bound is more often comprised of both an information and a linear relaxation, as discussed in the comparison of dual bounds above and in §6.2.1. In addition, however, the results in Table 2 show that this widening gap is due in large part to increased waiting and detouring time. Our routing policies have an estimate of the expected waiting time at extradepot CSs which increases with increasing demand. When seeking to avoid long expected queues, the routing policies perform longer detours, often back to the depot at which there is no queue. This also results in increased charging times for the non-anticipative policies. A particularly good example of lengthy detours is the high demand case for the TSP Static policy: it spends on average 10% of its time detouring, compared to an average of 6.9% for the other routing policies (and has the longest recharge times and worst objective performance as a result). We also note that the policy with PI does not always achieve the fastest average charge rate. Instead, to avoid waiting at CSs or performing lengthier detours, it will sacrifice fast charging, either by charging on slower segments of the charging function or by choosing a CS with slower charging technology. Lastly, to achieve a near-constant objective value with increasing demand, the optimal policy with PI is consistently able to find convenient extradepot CSs at which it incurs near-zero waiting times. The large gap between our policies and the PI bound emphasizes the value of this information.

Computational effort.

In Table 3, we report the computational effort for our policies and dual bounds. For the policies, we also include the average time required to make a decision in each epoch. For the PI bound and the routing policies, in general the better (lower) the objective value, the more computation time is required. As these results show, the 3% improvement of the optimal static policy over the TSP static policy and the 2% improvement of PreOpt over PostTSP come at a significant computational cost: more than eight days for the former and nine days for the latter. TSP Static's competitive objective achievements and relatively short computation time make it a good candidate for inclusion in more complex lookahead procedures, such as PostTSP. Here, we find that embedding TSP Static into a post-decision rollout improves performance by 0.7% while maintaining an average per-epoch computational effort of 1.4 s.

Interestingly, the time to compute the PI bound decreases with an increasing ratio of charging stations per customer (see Figure 10). This is likely due to the structure of the objective function in Equation (15). Recall that while the master problem (specifically, inequalities (23)-(29)) has approximations for the

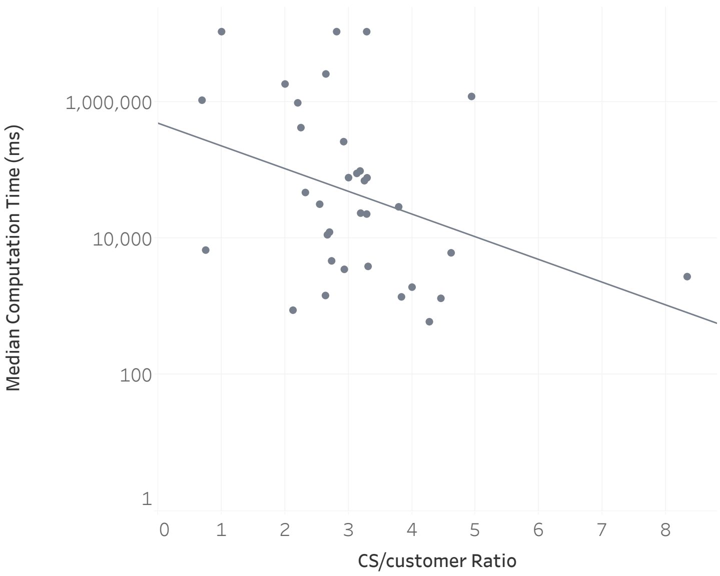

Figure 10: Computation times for instances' PI bounds versus their ratio of CSs to customers. *Note: the figure contains a point for each technician assignment, representing the sum over 50 realizations of uncertainty for each demand scenario.*

detouring and recharging time required to feasibly traverse a CL sequence, the exact amount – and any waiting time – is unknown and only revealed by solving the subproblem. As CSs become more abundant, more opportunities are available for low-cost detours and short waiting times, so the required amount of detouring and recharging time decreases. This improves the master problem's approximations of these values, consequently leading to faster solution times.

Disaggregated results over the testbed of instances are available as an appendix in §H.

7.3 Public-Private vs. Private-Only Recharging Strategies

Perhaps most importantly, we wish to demonstrate that even in the face of uncertainty at public charging stations our policies perform favorably relative to the private-only strategy. This is true by default a majority of the time, as the private-only strategy is energy-infeasible for 20/34 technician assignments. For the remaining instances, averaging across demand scenarios, we find that all proposed policies soundly outperform the best private-only solutions, with our best-performing policies outperforming the private-only solutions by 23%. See Table 4 and Figure 11. Together with the large number of infeasiblilities, our results suggest that committing to a private-only recharging strategy may lead to higher costs associated with the use of EVs, potentially hampering their adoption in commercial applications.

Of the 14 technician assignments for which the private-only strategy is feasible, we solve six to optimality and use the best solution found after three hours of computing time for the remaining eight. In general, we find that our route-based policies tend to outperform the best private-only solution by wide margins: 33.7% in low-demand scenarios, 25.5% in moderate-demand scenarios, and 9.7% in high-demand scenarios. This decrease in the gap with increasing demand is due to an increase in both waiting and detouring time, as in the comparison of policies to the PI bound. We also observe that the policies increasingly revert to routing behavior that relies on the depot for recharging, as expected waiting times at extradepot CSs exceed the additional time required to detour back to the depot. That is, the private-

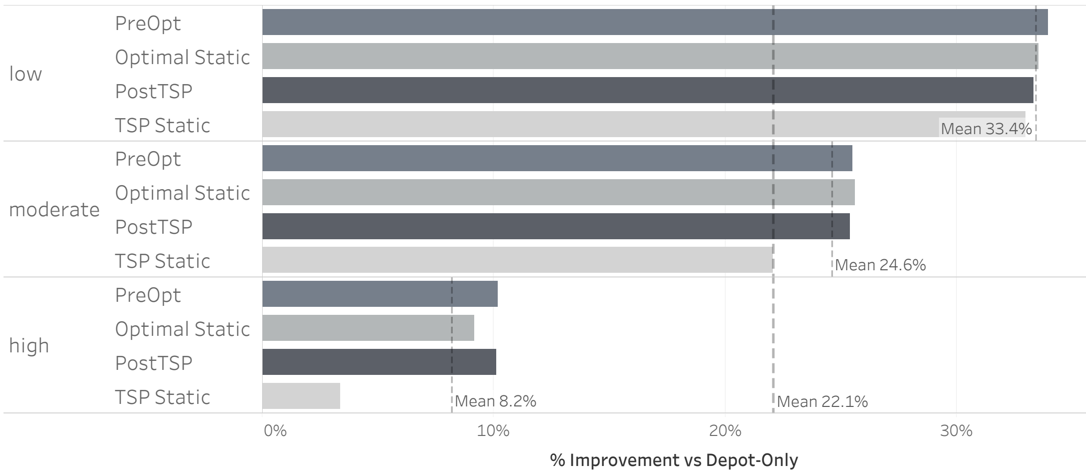

Figure 11: Depicting objective improvement of routing policies compared to the private-only solution. Dashed lines indicate mean values over demand levels. *Note: values are averages over the instances corresponding to the 14 technician assignments for which the private-only solution was feasible. Policies were evaluated on 50 realizations of uncertainty for each.*

only recharging approach is used as a fallback for public-private routing policies in some scenarios with high expected demand.

As EVs continue to increase in popularity and related technologies develop, it is likely that the performance gap between the private-only and public-private recharging strategies will widen further. Assuming charging infrastructure increases at a rate similar to its demand, the average performance of policies should improve. Intuitively, as more charging stations become available at which to charge, the detouring and waiting an EV must do prior to charging will reduce. Further, as more real-time information becomes available regarding demand at extradepot CSs, more informed routing decisions can be made as there will be less uncertainty, which should lead to better policy performance. We explore this possibility in §7.4.

Table 2: Detailed results comparing route-based policies to the value of the optimal policy with perfect information. *Note: Values for each demand scenario are averages over 34 technician assignments with 50 realizations of uncertainty for each.*

	low					moderate					high				
	PI	PreOpt	Opt Static	PostTSP	TSP Static	PI	PreOpt	Opt Static	PostTSP	TSP Static	PI	PreOpt	Opt Static	PostTSP	TSP Static
Objective	160.1	166.5	167.4	166.9	168.1	162.1	188.1	187.5	188.7	191.9	165.5	253.5	254.9	262.7	263.0
Objective (% diff from PI)	0.0	3.9	4.5	4.2	5.0	0.0	16.0	15.7	16.4	18.4	0.0	53.2	54.0	58.7	58.9
Detouring Time	2.6	3.7	2.7	3.6	3.4	3.4	10.3	9.3	6.6	6.4	7.2	19.1	18.6	15.3	26.0
Charging Time	13.2	13.8	13.5	14.0	14.1	13.1	15.0	14.6	15.1	14.8	12.7	17.6	17.3	18.9	20.2
Waiting Time	0.0	5.6	7.5	5.7	7.1	0.9	18.4	20.0	18.9	28.8	2.7	69.3	71.4	74.3	71.5
Charge Rate (kW)	39.8	39.8	39.6	39.6	38.8	39.7	41.8	41.9	41.6	39.2	39.0	41.8	41.8	40.9	39.7

Table 3: Computational effort for the routing policies (upper) as well as the establishment of dual bounds (lower). Left column is the aggregate over all computational experiments. Right column indicates the average time for policies to make a decision in each epoch. *Note: (1) The value for the CV bound is an aggregation over a single solution for each of the 34 technician assignments; values for the PI bound and all policies are aggregations over 50 realizations of uncertainty for each of the 102 instances. (2) Values for PreOpt include the time required to solve for the optimal static policy for the first epoch. In other words, if the optimal static fixed route was available in advance of running the PreOpt policy, then the computational effort would be the value shown for PreOpt minus the value for the optimal static policy.*

	Total computational effort (dd-hh:mm:ss)	Per-epoch computational effort (s)
PreOpt	11-14:49:19.8	9.0
Optimal Static	08-03:19:42.6	6.4
PostTSP	01-22:06:05.7	1.4
TSP Static	00-00:02:05.4	1.3
Myopic	00-00:00:07.0	0.1
PI Bound	205-00:18:53.0	-
CV Bound	00-00:00:01.8	-

Table 4: Comparing routing policies to solutions using private-only recharging strategy. *Note: Values are averages over the 14 technician assignments for which the private-only solution was feasible. Policies' reported performances are averages over 50 realizations of uncertainty for each demand scenario*

		low				moderate				high			
	Private-only	PreOpt	Opt Static	PostTSP	TSP Static	PreOpt	Opt Static	PostTSP	TSP Static	PreOpt	Opt Static	PostTSP	TSP Static
Objective	200.6	132.6	133.3	133.8	134.5	149.5	149.3	149.7	156.4	180.2	182.2	180.4	193.9
Direct-travel time (CL seq)	144.2	114.8	114.8	115.4	114.2	115.2	114.9	124.4	114.2	117.7	116.5	130.2	114.2
Detouring Time	34.8	5.4	4.1	5.3	5.2	16.9	16.6	7.7	9.0	23.7	23.5	12.1	36.8
Charging Time	21.7	7.7	7.2	8.0	7.8	9.3	9.1	9.4	9.0	12.0	11.7	12.4	16.0
Waiting Time	0.0	4.6	6.9	4.6	6.6	7.9	8.7	8.0	26.7	26.6	29.9	25.0	26.6

7.4 Impact of Future Technologies on Dynamic Policy Performance

Technology improvements may afford users of public CSs additional information regarding their availability. For example, real-time charging terminal data coupled with remote webcams could be used to determine the number of customers currently utilizing or waiting for a charger at a charging station. Access to this real-time information may be modeled as a relaxation for the current problem. That is, we can grant the decision maker in the E-VRP-PP access to the current state of the queue at each CS and assess policies' performance under this filtration; call the filtration $\mathbb{Q}$. The optimal performance of the decision maker under $\mathbb{Q}$ should be bounded below by the dual bounds and above by our current best-performing policy, since $\mathbb{F} \subseteq \mathbb{Q} \subseteq \mathbb{I}$. We explore the performance of our dynamic policies under $\mathbb{Q}$ here. Static policies, which leverage only long-term queue behavior and have fixed decision making, are unaffected by this change in available information.

7.4.1 Results.

We denote by *QPostTSP* the PostTSP policy under the $\mathbb{Q}$ filtration, and analogously *QPreOpt* for the PreOpt policy. For QPreOpt, we again initialize it with the route established by the Optimal Static policy. We then resolve for the optimal static policy using real-time queue information every time the vehicle reaches a CS with a longer-than-expected queue. While more frequent resolving would yield better objective performance, it also increases the computational burden of the QPreOpt policy, which is arguably already too long for use in practice (37.5 s per epoch, on average). Conversely, while QPostTSP requires more time to choose an action than PostTSP (4.1 s instead of 1.4 s), it is still sufficiently small to be executed at each epoch.

The performance of QPostTSP and QPreOpt are shown in relation to their counterparts under $\mathbb{F}$ (PostTSP and PreOpt, respectively) at left in Figure 12. We see that both policies have improved performance under $\mathbb{Q}$, as expected. However, this difference is significantly larger in the case of QPostTSP, outperforming PostTSP by 12.6% in high demand scenarios and 5.6% on average. In fact, in comparison to the PI bound (Figure 12, right), we see that QPostTSP outperforms both PreOpt and QPreOpt, with an overall average distance to the PI bound of 18.2%, making it our best performing policy. We also note that under the $\mathbb{Q}$ filtration, there is a substantial benefit to using dynamic policies over static policies. Whereas under $\mathbb{F}$ static and dynamic policies performed similarly on average (see Figure 9), the difference under $\mathbb{Q}$ is substantially larger. Our results for policy performance under the $\mathbb{Q}$ filtration suggest that adhering to the private-only recharging strategy will become increasingly costly as future technologies allow for more access to real-time information.

7.4.2 Modeling $\mathbb{Q}$.

To model the $\mathbb{Q}$ filtration, we modify the information space and the post-to-pre-decision transitions (see §2.2). The primary difference lies in the definition of w^z, the observation of the vehicle's position in queue. Rather than observing only its position at its current location ($w^z \in \mathbb{N}$), under $\mathbb{Q}$ it also receives observations of the number of vehicles at each extradepot CS ($w^z \in \mathbb{N}^C$). This information is then used to update its queue memories for each extradepot CS.

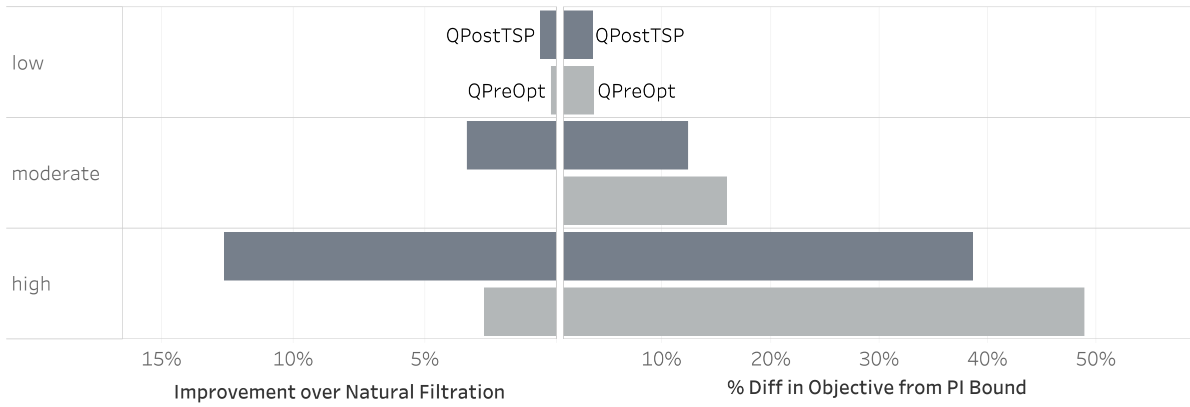

Figure 12: (Left) Relative objective improvement of routing policies under the $\mathbb{Q}$ filtration compared to the natural filtration. (Right) Relative performance of QPostTSP and QPreOpt compared to the PI bound (c.f., Figure 8). *Note: Each bar represents the average percent difference over all 34 technician assignments with 50 realizations of uncertainty each.*

8 Concluding Remarks

We have introduced the E-VRP with public-private recharging strategy and proposed an approximate dynamic programming solution. Through a decomposition of the E-VRP-PP, we bridge static and deterministic routing methods with dynamic and stochastic routing problems. Using these methods, we construct static and dynamic routing policies, including rollout algorithms and the optimal static policy. To better measure the goodness of these policies, we provide dual bounds. First, we provide a bound equal to the value of using a conventional vehicle. We then establish a tighter dual bound on the value of the optimal policy through the use of a perfect information relaxation. Using that dual bound, we have demonstrated that our routing policies are competitive with the optimal policy, coming within 25% on average and within 10% in the majority of instances.

Our work was motivated by an example from industry in which an EV operator rejected the public-private recharging strategy to avoid uncertainty at public charging stations. We sought to answer whether with a good dynamic routing policy such companies could adopt a public-private recharging strategy that would be cheaper than the private-only strategy. In computational experiments using real instances from industry, we found this to be true, demonstrating that all of our policies under the public-private recharging strategy soundly outperform the solution under a private-only strategy, with our best policies offering savings of approximately 23% on average. We further demonstrated that advancements in technology may make adhering to the private-only recharging strategy increasingly costly. Ultimately, we hope this work encourages companies to adopt a public-private recharging strategy, increasing the utility of EVs in commercial applications and accelerating the transition to sustainable transportation.

Acknowledgments

The authors especially want to thank Aurélien Froger for his assistance implementing the labeling algorithm and the Benders-based branch-and-bound. The authors also wish to thank David Brown for his valuable input on the implementation of information penalties. Nicholas Kullman is grateful for support from the Société des Professeurs Français et Francophones d'Amérique. Justin Goodson and

Jorge Mendoza extend their appreciation to the INFORMS TSL Society for their support of this research via the Cross-region Collaborative Grant. Justin Goodson also wishes to express appreciation for the support from the Center for Supply Chain Excellence at the Richard A. Chaifetz School of Business. This research was also partly funded by the French Agence Nationale de la Recherche through project e-VRO (ANR-15-CE22-0005-01).

References

Jonathan D Adler and Pitu B Mirchandani. Online routing and battery reservations for electric vehicles with swappable batteries. *Transportation Research Part B: Methodological*, 70:285–302, 2014.

Anders Reenberg Andersen. areenberg/mc_math: Create and evaluate queueing networks, March 2020. URL `https://doi.org/10.5281/zenodo.3725614`.

Anheuser-Busch. Anheuser-Busch Drives the Future of Beer Delivery with One of Largest Reported Pre-Orders of Tesla Electric Self-Driving Trucks, December 2017. `https://www.anheuser-busch.com/newsroom/2017/12/anheuser-busch-drives-the-future-of-beer-delivery-with-one-of-la.html`, Acceessed on April 13, 2020.

John Barco, Andres Guerra, Luis Muñoz, and Nicanor Quijano. Optimal routing and scheduling of charge for electric vehicles: A case study. *Mathematical Problems in Engineering*, 2017, 2017.

David B Brown, James E Smith, and Peng Sun. Information relaxations and duality in stochastic dynamic programs. *Operations research*, 58(4-part-1):785–801, 2010.

A. Campbell and B. Thomas. Challenges and advances in a priori routing. In B. Golden, S. Raghavan, and E. Wasil, editors, *The vehicle routing problem: latest advances and new challenges*. Springer, New York, NY, 2008.

George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.

E. Denardo. *Dynamic programming: models and applications*. Dover Publications, Mineola, NY, 2003.

Guy Desaulniers, Fausto Errico, Stefan Irnich, and Michael Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016.

Sevgi Erdoğan and Elise Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

Etalab. Fichier consolidé des bornes de recharge pour véhicules électriques, October 2014. `https://www.data.gouv.fr/en/datasets/fichier-consolide-des-bornes-de-recharge-pour-vehicules-electriques/`, Accessed on April 13, 2020.

FedEx. FedEx Introduces Zero-Emission All-Electric Nissan e-NV200 Vehicles in Belgium, April 2017. `https://about.van.fedex.com/newsroom/fedex-introduces-zero-emission-electric-nissan-e-nv200-vehicles-belgium/`, Accessed on April 13, 2020.

José Vicente Colomer Ferrándiz, María Amparo Saiz Gabaldón, and Óscar Colomer Font. The use of an electric vehicle fleet for the domiciliary hospitalization unit of the Hospital of Alcoy. *Transportation research procedia*, 18:411–418, 2016.

Aurélien Froger, Jorge E Mendoza, Ola Jabali, and Gilbert Laporte. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Computers & Operations Research*, 104:256–294, 2019.

Dominik Goeke and Michael Schneider. Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99, 2015.

Gerhard Hiermann, Jakob Puchinger, Stefan Ropke, and Richard F Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018, 2016.

Gerhard Hiermann, Richard F. Hartl, Jakob Puchinger, and Thibaut Vidal. Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1):235 – 248, 2019. ISSN 0377-2217.

Nicholas D Kullman, Aurelien Froger, Jorge E Mendoza, and Justin C Goodson. e-vro/frvcpy v0.1.0, February 2020. URL https://doi.org/10.5281/zenodo.3677583.

La Poste. La Poste et l'environnement, 2011. https://www.laposte.fr/legroupe/Nos-engagements/Developpement-durable/La-Poste-et-l-environnement%20?, Accessed on April 13, 2020.

Jorge E Mendoza, C Guéret, M Hoskins, H Lobit, V Pillac, T Vidal, and D Vigo. VRP-REP: the vehicle routing community repository. In *Third Meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog). Oslo, Norway*, 2014.

Alejandro Montoya, Christelle Guéret, Jorge E Mendoza, and Juan G Villegas. The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103:87–110, 2017.

Patrick Morrissey, Peter Weldon, and Margaret O'Mahony. Future standard and fast charging infrastructure planning: An analysis of electric vehicle charging behaviour. *Energy Policy*, 89:257–270, 2016.

A Omidvar and R Tavakkoli-Moghaddam. Sustainable vehicle routing: Strategies for congestion management and refueling scheduling. In *Energy Conference and Exhibition (ENERGYCON), 2012 IEEE International*, pages 1089–1094. IEEE, 2012.

Orlando Utilities Commission. Championing the EV charge, 2018. https://www.ouc.com/environment-community/green-initiatives/championing-the-ev-charge, Accessed on September 4, 2019.

Samuel Pelletier, Ola Jabali, and Gilbert Laporte. 50th anniversary invited article - goods distribution with electric vehicles: review and research perspectives. *Transportation Science*, 50(1):3–22, 2016.

Harilaos N Psaraftis, Min Wen, and Christos A Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31, 2016.

Maximilian Schiffer, Michael Schneider, and Gilbert Laporte. Designing sustainable mid-haul logistics networks with intra-route multi-resource facilities. *European Journal of Operational Research*, 265(2):517 – 532, 2018. ISSN 0377-2217.

Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014.

Timothy M Sweda, Irina S Dolinskaya, and Diego Klabjan. Adaptive routing and recharging policies for electric vehicles. *Transportation Science*, 51(4):1326–1348, 2017.

Tesla. Service, 2018. https://www.tesla.com/service, Accessed on April 13, 2020.

UPS. UPS To Deploy New, State-Of-The-Art Electric Vehicles In London And Paris, May 2018. https://pressroom.ups.com/pressroom/ContentDetailsViewer.page?ConceptType=PressReleases&id=1525867012405-924, Accessed on April 13, 2020.

Juan G. Villegas, Christelle Guéret, Jorge E. Mendoza, and Alejandro Montoya. The technician routing and scheduling problem with conventional and electric vehicle. working paper, June 2018. URL https://hal.archives-ouvertes.fr/hal-01813887.

A Defining the set of fixed routes

To define the set P of all possible fixed routes from a state $s_{k'}$, we first define the modified action space $\mathcal{A}^-(s_k)$, which allows the EV to start charging immediately regardless of queue length:

$$\begin{aligned}
\mathcal{A}^-(s_k) = \Bigg\{ & (a^i, a^q) \in \{\bar{\mathcal{N}}_k \cup \mathcal{C}\} \times [0, Q] : \\
& a^i = i_k, a^q \in \Bigg\{ \tilde{q} \in \mathcal{Q} \Bigg| \tilde{q} > q_k \wedge \\
& \big(\big(\exists c \in \mathcal{C},\, \exists j \in \bar{\mathcal{N}}_k :\ \tilde{q} \geq e_{i_k j} + e_{jc}\big) \vee \big(\bar{\mathcal{N}}_k = \emptyset \wedge \tilde{q} \geq e_{i_k 0}\big)\big) \Bigg\}, \\
& \quad i_k \in \mathcal{C} && (35) \\
& a^i \in \bar{\mathcal{N}}_k, a^q = q_k - e_{i_k a^i}, \\
& \quad (\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) && (36) \\
& a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i}, \\
& \quad q_k \geq e_{i_k a^i} \Bigg\}. && (37)
\end{aligned}$$

In contrast to the definition of $\mathcal{A}(s_k)$, there is no waiting action (c.f., equation (2)), and we remove the condition from equation (35) that requires a charger be available for the vehicle to recharge. In addition, we define $\mathcal{S}^-(s_k, a)$ to be the set of reachable states in epoch $k+1$ when choosing action a from state s_k and when the exogenous information observed is $W_{k+1} \in \{(w^t, w^z)|(w^t, w^z) \in \mathcal{I}(s_k^a) \wedge w^z = 1\}$ (effectively, we ignore any information regarding position in queue, assuming it is 1 everywhere we go). Then we may define the set P of all fixed routes from a state $s_{k'}$ recursively as follows:

$$P = \{(p_1, p_2, \ldots, p_D)|p_j \in P_j,\, 1 \leq j \leq D\},$$

where D is the (variable) index of the terminal direction and the P_js are the sets of possible directions available $j-1$ steps into the future, defined as

$$\begin{aligned}
P_1 &= \{(i_{k'}, q_{k'})\} \\
P_2 &= \mathcal{A}^-(s_{k'}) \\
&\vdots \\
P_j &= \bigcup_{s' \in \mathcal{S}^-(s_{(k'+j-3)}, p_{j-1})} \mathcal{A}^-(s') \\
&\vdots \\
P_D &= \{(0, q)|q \in [0, Q]\}.
\end{aligned}$$

B Proof of Proposition 1

We begin by repeating the statement for Proposition 1:

For all static, non-AC-policies $\pi \in \Pi^B$, there exists an AC policy $\pi^{AC} \in \Pi^{AC}$ whose objective value is no worse: $\tau(\pi^{AC}) \leq \tau(\pi)$.

Proof. Proof. In order for a policy $\pi \in \Pi^{\mathrm{B}}$ to be non-AC, it must visit CSs without charging at them. We refer to this as "balking" a CS. Consider a vehicle operating under the static non-AC policy π which balks CSs. We wish to show that there exists a static AC-policy π^{AC} such that $\tau(\pi^{\mathrm{AC}}) \leq \tau(\pi)$. We can trivially construct such a policy by simply mimicking π, except when π balks a CS. In that case, the constructed policy π^{AC} would skip visiting the balked CS and proceed directly to the subsequent location. For instance, if the static policy π dictates the relocation from some node j to a charging station c and then immediately relocate to j', policy π^{AC} would proceed directly from j to j'. In so doing, the objective value of policy π^{AC} will differ from that of π by an amount $t_{jc} + t_{cj'} - t_{jj'}$. Because the triangle inequality holds for travel times and queues are served first-in-first-out (FIFO), this policy will have expected cost no larger than that of π. □ □

The intuition is that because static policies follow a predetermined set of actions, visiting a charging station without the intent to charge serves no purpose except to increase the time required to complete the route. In the case of dynamic policies, they may visit a charging station and ultimately balk, but this would be in response to the observation of the queue length at the charging station, rather than a premeditated immediate departure. The construction strategy for π^{AC} in the proof requires knowledge of these immediate departures a priori, so it is therefore only valid in the context of static policies. We note that this proof holds under any information filtration.

C Proof of Proposition 2

We begin by repeating the statement for Proposition 2:

For AC policies beginning in a state s_k, the E-VRP-PP can be decomposed into routing and charging decisions with objective

$$\min_{\pi(p)\in\Pi^{AC}} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi(p)}(s_{k'}))\right] = \min_{\rho\in\mathcal{R}(s_k)} \left\{ \min_{\pi\in\Pi_\rho} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi}(s_{k'}))\right]\right\}.$$

Proof. Proof. Because each AC policy $\pi(p) \in \Pi^{\mathrm{AC}}$ maps to a CL sequence $r(\pi(p))$ given by equation (14), we may equivalently write the set of AC policies as $\Pi^{\mathrm{AC}} = \bigcup_{\rho\in\mathcal{R}(s_k)} \Pi_\rho$, where $\Pi_\rho = \{\pi(p) \in \Pi^{\mathrm{AC}} : r(\pi(p)) = \rho\}$. This partitioning of the policy set allows us to write the objective function as a nested minimization over CL sequences and their corresponding fixed-route policies:

$$\min_{\pi(p)\in\Pi^{\mathrm{AC}}} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi(p)}(s_{k'}))\right] = \min_{\rho\in\mathcal{R}(s_k)} \left\{ \min_{\pi\in\Pi_\rho} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi}(s_{k'}))\right]\right\}. \quad \square$$

□

D Defining Action Space for FRVCPs

FRVCPs may be modeled as dynamic programs with formulations identical to the primary formulation for the E-VRP-PP outlined in §2, except the FRVCP operates under a more restricted action space $\mathcal{A}^{\mathrm{AC}}(s_k, \rho)$. Here we offer a formal definition of $\mathcal{A}^{\mathrm{AC}}(s_k, \rho)$. This action space disallows non-AC policies, and it ensures that the vehicle follows the CL sequence ρ. Let $\bar{\mathcal{N}}_k' = \bar{\mathcal{N}}_k \cup \{0\}$, and define the function $n : (\mathcal{R} \times \mathcal{S}) \to \bar{\mathcal{N}}_k'$ which maps a CL sequence ρ and state s_k to the next element in ρ to be visited. For simplicity, we call this element $n^\star = n(\rho, s_k)$. Then we define $\mathcal{A}^{\mathrm{AC}}(s_k, \rho)$ by the following:

$$\mathcal{A}^{\mathrm{AC}}(s_k,\rho) = \Big\{ (a^i, a^q) \in \{n^\star \cup \mathcal{C}\} \times [0, Q] :$$

$$a^i = i_k, a^q = q_k, \quad i_k \in \mathcal{C}' \wedge \psi_{i_k} < z_k \tag{38}$$

$$a^i = i_k, a^q \in \{\tilde{q} \in \mathcal{Q} | \tilde{q} > q_k \wedge (\exists c \in \mathcal{C} : \tilde{q} \geq e_{i_k n^\star} + e_{n^\star c})\}, \quad i_k \in \mathcal{C} \wedge z_k \leq \psi_{i_k} \tag{39}$$

$$a^i = n^\star, a^q = q_k - e_{i_k a^i}, \quad (\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) \wedge (i_k \in C \Rightarrow q_k > q_{k-1}) \tag{40}$$

$$a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i}, \quad q_k \geq e_{i_k a^i} \wedge (i_k \in C \Rightarrow q_k > q_{k-1}) \Big\}. \tag{41}$$

The action space $\mathcal{A}^{\mathrm{AC}}(s_k, \rho)$ is identical to $\mathcal{A}(s_k)$ with the following exceptions. First, it contains the additional condition $i_k \in C \Rightarrow q_k > q_{k-1}$ in equations (40) and (41). This condition specifies that the vehicle may only depart a CS if it charged in the previous epoch. Second, we require $a^i = n^\star$ in equation (40). This ensures that, when deciding to visit a customer, it is the next one in the CL sequence ρ. Finally, we modify the set of charging decisions in equation (39) such that the vehicle always charges to an energy level sufficient to reach the next location $n^\star$.

E Modifications to Froger et al. (2019) algorithm for the FRVCP

Froger et al. (2019) propose an exact algorithm to solve the FRVCP when the charging functions are concave and piecewise-linear and the charging decisions are continuous. In their implementation, waiting times at charging stations are not considered. We modify the algorithm to accommodate discrete charging decisions and time-dependent waiting times at the charging stations. For this discussion, additional information about the algorithm beyond the overview in §4.4 is necessary. We refer the reader to the description of Algorithm 3 in Froger et al. (2019), which is primarily located in their §5.3 and Appendix E.

To handle discrete charging decisions, we first modify the set of breakpoints that define the charging functions. Namely, we include a "breakpoint" in the charging function at each $q' \in \mathcal{Q}$ (even if the slope of the charging function does not change at q'). Next, we modify the process of extending a label. Consider the example of the edge connecting nodes 1 and nodes 4d in the graph $\mathcal{G}'$ in Figure 5. During the translation of the SoC function by $(t_{14}, -e_{14})$, as in the original implementation, we remove all resulting supporting points with negative SoC. However, in the original implementation in which charging decisions were continuous, a new supporting point was added at the translated SoC function's intersection with the x-axis. This allowed the vehicle to charge just enough at the previous CS to be able to reach the new node with zero energy. With discrete charging decisions, we no longer create this point, so the SoC function for the label at node 4d has only three supporting points: $\{\tilde{z}_1, \tilde{z}_2, \tilde{z}_3\}$. See Figure 13.

To accommodate time-dependent waiting times, we make additional adjustments to the SoC function when extending a label to a CS node, such as to node 4d. We want the supporting points in the SoC function to reflect the time at which the vehicle enters service at the CS. To do so, after the initial translation (depicted in Figure 13), we shift the SoC function supporting points again according to the underlying wait time (either known or expected). Define the function $w : (\mathbb{R}_{\geq 0} \times \mathcal{C}) \rightarrow \mathbb{R}_{\geq 0}$ that specifies

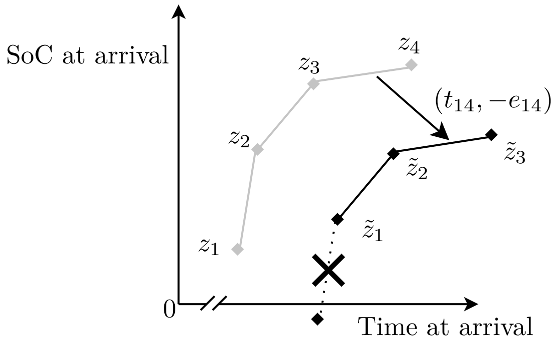

Figure 13: An example of shifting the SoC function as we extend the label along the edge from customer node 1 to CS node 4d in Figure 5. The SoC function for node 1 (in gray) is translated by $(t_{14}, -e_{14})$. The resulting SoC function for the label at node 4d (in black) contains one fewer supporting point, since the translation of z_1 yields an infeasible point with negative SoC.

the (known or expected) waiting time that the EV incurs if it arrives to some CS c at some time t. In the case of known waiting times, functions $w(t, c')$ may not be continuous for a given CS c' (see Figure 4, right), so we cannot represent the resulting SoC function as continuous. We group the supporting points based on discontinuities in $w(t, c')$ and create a new label for each group.

For example, consider again extending the label from customer node 1 to CS node 4d in Figure 5. After the initial shift of the SoC function, we are left with the supporting points $\{\tilde{z}_1, \tilde{z}_2, \tilde{z}_3\}$ shown in black in Figure 13. Now, in Figure 14 we consider known, time-dependent waiting times. The underlying wait-time function $w(t, 4)$ (top graph, in gray) has a discontinuity at the time $t = \phi$ between supporting points $\tilde{z}_2$ and $\tilde{z}_3$. As a result, the supporting points are split into two groups ($\{\tilde{z}_1, \tilde{z}_2\}$ and $\{\tilde{z}_3\}$, shown in bottom graphs) each of which comprises a new label. All supporting points $\tilde{z}_j$ are then shifted by the amount $w(\tilde{z}^t_j, 4)$ to produce the final SoC functions for these labels.

Figure 14 depicts an example for the FRVCP-P, in which wait time functions may not be continuous. In contrast, wait-time functions for the FRVCP-N are always continuous (see Figure 4, left), so there is no need to divide the supporting points and create multiple labels. We simply shift each supporting point by its underlying wait time.

F Information Penalties

The dual bound achieved with perfect information (see §6.2) is often loose, because no decision maker is clairvoyant and advanced knowledge of the future is often valuable. To tighten the bound, we can penalize the decision maker and attempt to eliminate any benefit of using advanced information. These information penalties manifest as additional costs $z(s_k, a)$ incurred during action selection in the perfect information problem. We write the objective function of the penalized perfect information problem as

$$\mathbb{E}\left[\min_{\pi\in\Pi}\sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) + z\left(s_k, X_k^\pi(s_k)\right)\middle| s_0\right]. \tag{42}$$

The form of the information penalty we use is $z(s_k, a) = \mathbb{E}\left[V_{k+1}(s_k, a)|\mathscr{F}_k\right] - \mathbb{E}\left[V_{k+1}(s_k, a)|\mathscr{I}_k\right]$, where $V_{k+1}(s_k, a)$ is the value of being in the pre-decision state s_{k+1} reached by choosing action a from state s_k. The penalty captures the difference in the expected cost-to-go under the natural and perfect

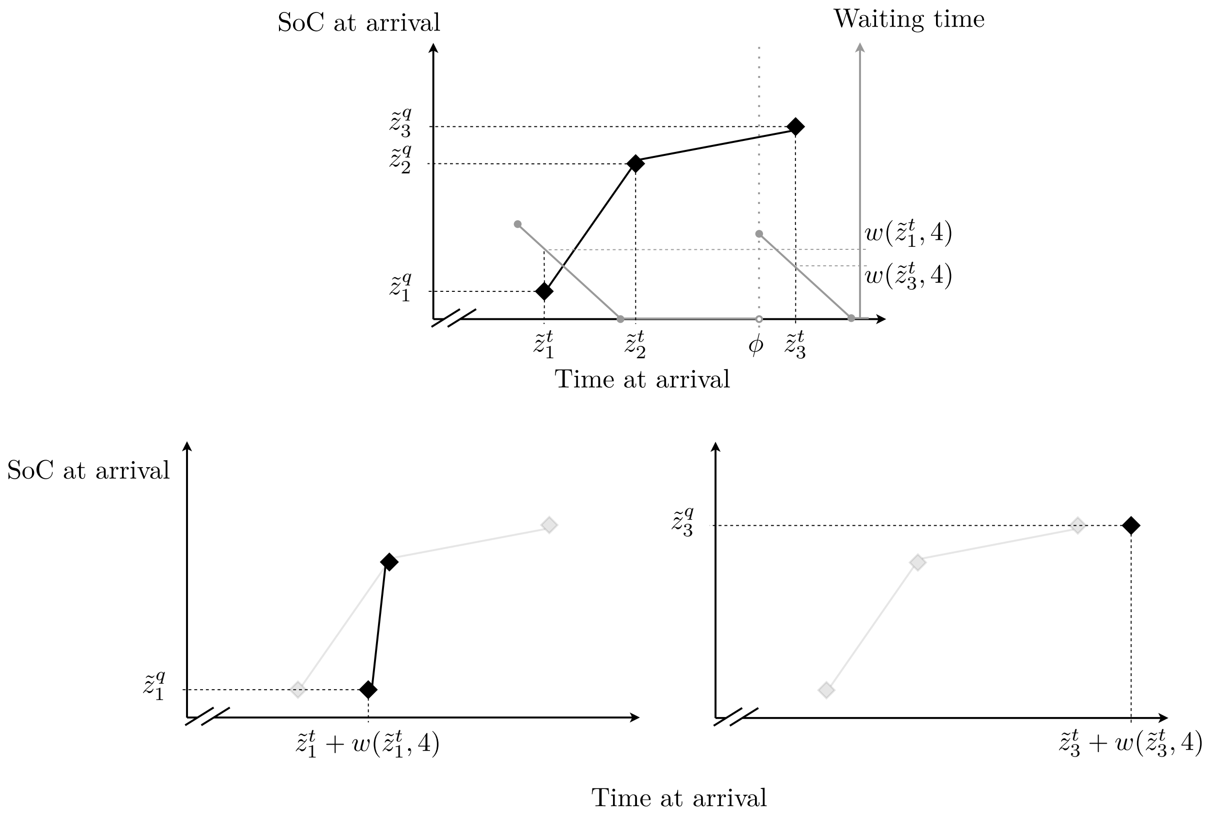

Figure 14: Depiction of handling time-dependent waiting times. In the top graph, we have the resulting SoC function after the initial translation from node 1 to node 4d depicted in Figure 13. This is superimposed over the wait-time function $w(t, 4)$, plotted in gray. The supporting points for the SoC function are divided into groups on either side of the discontinuity at $t = \phi$, resulting in two new labels shown in the bottom two graphs. After this division, the SoC functions' supporting points are shifted by their wait times. The final SoC functions are shown in black, superimposed over the pre-divided, pre-shifted SoC function.

information filtrations. The form of this penalty aligns with that of Theorem 2.3 (and Proposition 2.2) of Brown et al. (2010), which promises strong duality. Strong duality guarantees that the optimal objective value of the penalized perfect information problem (42) will be equal to the objective value of the optimal non-anticipative policy. In practice, however, the values $\mathbb{E}\left[V_{k+1}(s_k, a) | \mathscr{F}_k\right]$ and $\mathbb{E}\left[V_{k+1}(s_k, a) | \mathscr{I}_k\right]$ are unknown. To approximate them, we follow an approach suggested in Brown et al. (2010), employing value function approximations for $V_{k+1}(s_k, a)$.

Let $v^{\mathbb{G}}_{k+1}(s_k, a)$ be the approximation of $\mathbb{E}\left[V_{k+1}(s_k, a) | \mathscr{G}_k\right]$ under a filtration $\mathbb{G}$. Then we can write our approximated penalty as $\hat{z}(s_k, a) = v^{\mathbb{F}}_{k+1}(s_k, a) - v^{\mathbb{I}}_{k+1}(s_k, a)$. To compute $v^{\mathbb{G}}_{k+1}(s_k, a)$ we utilize an *estimating policy* $\pi(s_{k+1}, \mathbb{G})$ to approximate the cost-to-go from a future state s_{k+1} under the filtration $\mathbb{G}$: $v^{\mathbb{G}}_{k+1}(s_k, a) = \mathbb{E}\left[\sum_{i=k+1}^{K} C\left(s_i, X_i^{\pi(s_{k+1}, \mathbb{G})}(s_i)\right) \middle| s_k, a\right]$. For our estimating policy, we use the TSP static policy (see §5.1.2). Then we may write our penalty explicitly as

$$\hat{z}(s_k, a) = \mathbb{E}\left[\sum_{i=k+1}^{K} C\left(s_i, X_i^{\pi^{\mathrm{TSP}}(s_{k+1}, \mathbb{F})}(s_i)\right) \middle| s_k, a\right] - \mathbb{E}\left[\sum_{i=k+1}^{K} C\left(s_i, X_i^{\pi^{\mathrm{TSP}}(s_{k+1}, \mathbb{I})}(s_i)\right) \middle| s_k, a\right] \tag{43}$$

The objective for the penalized PI problem with our approximation is

$$\begin{aligned}
&\mathbb{E}\left[\min_{\pi \in \Pi} \sum_{k=0}^{K} C\left(s_k, X_k^{\pi}(s_k)\right) + \hat{z}\left(s_k, X_k^{\pi}(s_k)\right) \middle| s_0\right] \\
&= \mathbb{E}\left[\min_{\pi \in \Pi} \sum_{k=0}^{K} C\left(s_k, X_k^{\pi}(s_k)\right) + v^{\mathbb{F}}_{k+1}\left(s_k, X_k^{\pi}(s_k)\right) - v^{\mathbb{I}}_{k+1}\left(s_k, X_k^{\pi}(s_k)\right) \middle| s_0\right].
\end{aligned} \tag{44}$$

As in the unpenalized perfect information problem, without loss of optimality, we may restrict our search of policies to those that are AC. We justify this restriction in Proposition 4.

Proposition 4 (Optimal policies for penalized PI problem are AC)**.** *Let $\tau_{\hat{z}}(\pi)$ be the value of a policy $\pi \in \Pi$ for the penalized perfect information problem* (42)*, where the penalty is $\hat{z}$ as defined in equation* (43)*. Then for any non-AC policy $\pi \in \Pi^B$, there exists an AC policy $\pi^{AC} \in \Pi^{AC}$ such that $\tau_{\hat{z}}(\pi^{AC}) \leq \tau_{\hat{z}}(\pi)$.*

Proof. Proof. See §F.2. □

Following from Propositions 2 and 4, we may decompose the penalized perfect information problem into routing and charging decisions as before, so the objective function becomes

$$\mathbb{E}\left[\min_{\rho \in \mathcal{R}(s_0)} \left\{\min_{\pi \in \Pi_\rho} \sum_{k=0}^{K} C\left(s_k, X_k^{\pi}(s_k)\right) + v^{\mathbb{F}}_{k+1}\left(s_k, X_k^{\pi}(s_k)\right) - v^{\mathbb{I}}_{k+1}\left(s_k, X_k^{\pi}(s_k)\right)\right\} \middle| s_0\right]. \tag{45}$$

We can again estimate the objective value of (45) using simulation, as we did to estimate the unpenalized objective value with perfect information in (34). The inner minimization of (45) is still an FRVCP which can be modeled as a modified version of our original dynamic program, as in §4.3.2. To solve the penalized FRVCP, we use the classical reaching algorithm (Denardo 2003) that enumerates in forward-DP fashion all states that can be realized along a fixed CL sequence ρ. The restriction to AC policies in Proposition 4 is crucial, as it significantly reduces the number of realizable states that must be enumerated in the reaching algorithm. We also remove queue memories from the state description to further reduce the number of enumerable states. While time-consuming, the reaching algorithm allows for the consideration of nonlinear penalties, which can no longer be accommodated by the labeling algorithm nor by more classical solution methods, such as mixed integer-linear programs.

For an example of the construction of information penalties, let us consider Figure 15 with the vehicle in state $s_1 = (t_{0,3}, 3, Q - e_{0,3}, \{2, 1\}, 1)$ (note the omission of queue memories). We assume CSs 4 and

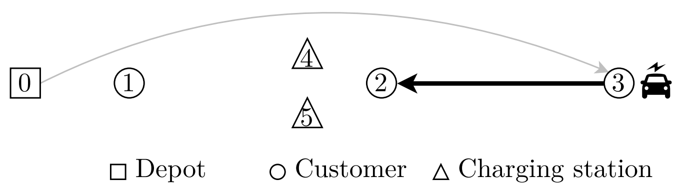

Figure 15: A vehicle at customer 3 in the beginning of epoch one. The vehicle must visit customers 2 and 1 before returning to the depot, but before it can visit customer 1, it must first charge. We illustrate the construction of information penalties using the action associated with the bolded arrow as an example.

5 are identical, meaning they have the same charging technology and number of chargers. Further, we assume that $t_{2,4} = t_{2,5}$ and $t_{4,1} = t_{5,1}$ (likewise for the energy to traverse these arcs). We compute a penalty for each action in the action space $\mathcal{A}(s_1)$, which consists of relocation actions to customer 2 and charging stations 4 and 5 (relocating to nodes 0 and 1 is energy infeasible). Abusing notation slightly, we have $\mathcal{A}(s_1) = \{a_2 \equiv (2, q_1 - e_{3,2}); a_4 \equiv (4, q_1 - 3_{3,4}); a_5 \equiv (5, q_1 - e_{3,5})\}$. In this example, we will illustrate the computation of the penalty $\hat{z}(s_1, a_2)$ corresponding to the action a_2 in which the EV relocates to customer 2.

First, from the post-decision state $s_1^{a_2}$, we sample realizations of queue dynamics at CSs 4 and 5. For simplicity, let us assume we are conducting a single sample denoted by $\omega \in \Omega$. We realize the (deterministic) exogenous information $W_2 = (t_1 + t_{3,2}, 1) \in \mathcal{I}(s_1^{a_2})$ and transition to state $s_2 = (t_{0,3} + t_{3,2}, 2, Q - e_{0,3} - e_{3,2}, \{1\}, 1)$. From this state, we wish to construct TSP Static policies $\pi(s_2, \mathbb{F})$ and $\pi(s_2, \mathbb{I})$ for use in $v_2^{\mathbb{F}}(s_1, a_2)$ and $v_2^{\mathbb{I}}(s_1, a_2)$, respectively. Per §5.1.2, the CL sequence followed by the vehicle will be the same under both filtrations, so we determine it first. To do so, we solve a single iteration of the outer minimization of equation (15). This finds the shortest Hamiltonian path from customer 2, through the remaining customers, terminating at the depot, which is the sequence $\rho = (2, 1, 0)$. Then, given ρ, we solve a single iteration of the inner minimization to establish the fixed route for the TSP static policies: we solve the FRVCP-N on ρ to construct the fixed route we call $p^{\mathbb{F}}$ and its corresponding policy $\pi(s_2, \mathbb{F}) = \pi(p^{\mathbb{F}})$, and we solve the FRVCP-P on ρ to construct the fixed route we call $p^{\mathbb{I}}$ with corresponding policy $\pi(s_2, \mathbb{I}) = \pi(p^{\mathbb{I}})$. For the former, let us assume that the expected waiting time at CS 4 is 40 min, and the expected waiting time at CS 5 is 45 min. This leads to the fixed-route solution $p^{\mathbb{F}} = ((2, q_2), (4, q_2 - e_{2,4}), (4, \tilde{q}), (1, \tilde{q} - e_{4,1}), (0, \tilde{q} - e_{4,1} - e_{1,0}))$, which includes a stop to charge at CS 4 to charge level $\tilde{q} = \min\{q \in \mathcal{Q}'\}$ where $\mathcal{Q}' = \{q \in \mathcal{Q} : q \geq e_{4,1} + e_{1,0}\}$. The cost of $p^{\mathbb{F}}$ we denote $\tau(\pi(p^{\mathbb{F}})) = t_{2,4} + 40 + \bar{u}(q_2 - e_{2,4}, \tilde{q}) + t_{4,1} + t_{1,0}$. For the FRVCP-P we proceed similarly, except now we have access to ω, which grants us knowledge of the queue dynamics at CS 4 and 5 at all points in time. Say we know the wait time at CS 4 will actually be 20 min, and the wait time at CS 5 will be 5 min. Then the solution to the FRVCP-P is the fixed route $p^{\mathbb{I}} = ((2, q_2), (5, q_2 - e_{2,5}), (5, \tilde{q}), (1, \tilde{q} - e_{5,1}), (0, \tilde{q} - e_{5,1} - e_{1,0}))$ with corresponding cost $\tau(\pi(p^{\mathbb{I}})) = t_{2,5} + 5 + \bar{u}(q_2 - e_{2,5}, \tilde{q}) + t_{5,1} + t_{1,0}$. The values $v_2^{\mathbb{F}}(s_1, a_2)$ and $v_2^{\mathbb{I}}(s_1, a_2)$ are then equal to the average of the route costs associated with $\pi(p^{\mathbb{F}})$ and $\pi(p^{\mathbb{I}})$, respectively, over samples from Ω (of which there is only one in this example). Thus, we have $\hat{z}(s_1, a_2) = v_2^{\mathbb{F}}(s_1, a_2) - v_2^{\mathbb{I}}(s_1, a_2) = \mathbb{E}\left[\tau(\pi(p^{\mathbb{F}}))\right] - \mathbb{E}\left[\tau(\pi(p^{\mathbb{I}}))\right] = 40 - 5 = 35$, so the penalized cost of choosing action a_2 from state s_1 is $C(s_1, a_2) + \hat{z}(s_1, a_2) = t_{3,2} + 35$. The value of the penalty represents the benefit of using advanced information in decision making, capturing the difference in expected costs-to-go $\mathbb{E}\left[V_2(s_1, a_2) | \mathscr{F}_1\right]$ and $\mathbb{E}\left[V_2(s_1, a_2) | \mathscr{I}_1\right]$.

While the CL sequence $\rho = (2, 1, 0)$ will be the same for each sample from Ω, the same is not generally true of $p^{\mathbb{I}}$ and $p^{\mathbb{F}}$, which must be resolved for each sample of queue dynamics. This process is repeated for each action in the action space and at each decision epoch.

As the example illustrates, the application of information penalties increases computation significantly, which restricts the size of instances in which we can apply them. This exercise may not be not in vain, however, as methods that yield near-optimal policies for smaller instances may portend toward good methods for larger instances.

F.1 Experiments with Information Penalties

To demonstrate the utility of information penalties we seek instances for which access to perfect information is exceptionally valuable. These instances should result in a large gap between the performance of a non-anticipative policy and one with perfect information, making for a weak dual bound. Good information penalties should then tighten the dual bound, demonstrating that our policies are closer to the optimal policy than originally suggested by the PI bound. We attempt to construct such an instance here by 1) including "competing" charging stations between which the EV must choose, and 2) increasing the amount of stochastic costs (waiting costs) relative to deterministic costs (traveling and charging costs). The former produces more uncertainty and a larger action space, both of which stand to increase the value of perfect information. The latter aims to simply highlight this value.

Because the reaching algorithm used to solve the penalized FRVCP (the inner minimization of (45)) enumerates all reachable states along a fixed CL sequence, we must be mindful of instance size in these experiments. To ensure tractability, we construct an instance with four customers and two extradepot CSs. Further, we limit the set of chargeable battery states $\mathcal{Q}$ to the charging function breakpoints and multiples of 25% ($\mathcal{Q} = \{0, 0.25Q, 0.5Q, 0.75Q, 0.85Q, 0.95Q, Q\}$). Despite these restrictions, the computational effort required to solve just one realization of uncertainty with information penalties is almost ten minutes. This is in contrast to the negligible computation time (milliseconds) required to establish the perfect information bound for this instance, as well as execute all other routing policies.

The experimental results for this instance over 250 samples of uncertainty are shown in Figure 16. The figure shows the performance of the optimal policy with perfect information ("PI"), the optimal policy with penalized access to perfect information ("PI + Penalty"), and our best dynamic and static policies (PostTSP and the optimal static policy, respectively). The size of the gap between our best policy and the PI bound (15.3%) suggests that we were successful in creating an instance in which information was valuable. The penalties' potential is evident in these results, as they yield a dual bound that is more than twice as strong: the gap between our best non-anticipative policy and the dual bound is 7.6% with penalties, compared to 15.3% with the PI bound alone.

To the best of our knowledge, these experiments represent the first successful demonstration of information penalties in vehicle routing and the first successful application of information penalties in general to a combinatorial perfect information problem lacking any special structure making the problem easier to solve. While scalability remains an issue, we hope that this serves as a proof-of-concept for future endeavors from other researchers.

F.2 Proof of Proposition 4

We begin by repeating the statement of Proposition 4:

Let $\tau_{\hat{z}}(\pi)$ *be the value of a policy* $\pi \in \Pi$ *for the penalized perfect information problem* (42)*, where the penalty is* $\hat{z}$ *as defined in equation* (43)*. Then for any non-AC policy* $\pi \in \Pi^B$*, there exists an AC policy* $\pi^{AC} \in \Pi^{AC}$ *such that* $\tau_{\hat{z}}(\pi^{AC}) \leq \tau_{\hat{z}}(\pi)$*.*

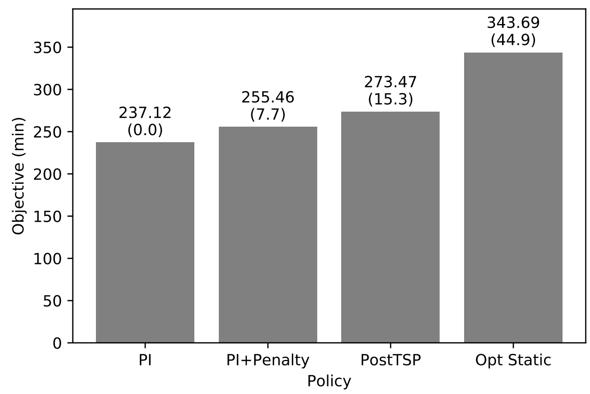

Figure 16: Comparing our best dynamic and static non-anticipative policies to the dual bounds afforded by the value of the optimal policy with perfect information and the value of the optimal policy with penalized access to perfect information. *Note: Bar labels are average objective achievement over 250 samples of uncertainty with percent difference from the PI bound in parentheses.*

Proof. Proof. We proceed similarly as in the proof of Proposition 1. Consider a vehicle operating under the non-AC policy π which balks CSs. We wish to show that there exists an AC policy π^{AC} such that $\tau_{\hat{z}}(\pi^{\text{AC}}) \leq \tau_{\hat{z}}(\pi)$. We can construct such a policy by mimicking π, except when π balks a CS. In that case, the constructed policy π^{AC} would skip visiting the balked CS and proceed directly to the subsequent location. For instance, if the policy π dictates the relocation from some node j to a charging station c and then immediately relocate to j', policy π^{AC} would proceed directly from j to j'.

In the proof of Proposition 1, we relied on the triangle inequality and the fact that our queues are served first-in-first-out to reason that the constructed policy π^{AC} would outperform π. Now in the presence of penalties, while the FIFO principle still holds, it is less obvious that the triangle inequality holds. We prove here that it does by comparing the costs and penalties incurred between j and j' under policies π^{AC} and π. More specifically, we want to show that

$$t_{j,j'} + \hat{z}(s_j, a_{j,j'}) \leq t_{j,c} + \hat{z}(s_j, a_{j,c}) + t_{c,j'} + \hat{z}(s_c, a_{c,j'}),$$

where s_j is the initial state of the vehicle at j; $a_{j,j'}$ is the action of traveling directly from j to j'; $a_{j,c}$ is the action of traveling from j to c; s_c is the state of the vehicle after taking action $a_{j,c}$ from state s_j; and $a_{c,j'}$ is the action of traveling from c to j'. The left-hand side of the equation represents the costs associated with traveling directly from j to j' (π^{AC}) and the right-hand side represents the costs associated with traveling from j to c, balking at c, then traveling to j' (π).

By the unpenalized triangle inequality, $t_{j,j'} \leq t_{j,c} + t_{c,j'}$, so it is sufficient to show that

$$\hat{z}(s_j, a_{j,j'}) \leq \hat{z}(s_j, a_{j,c}) + \hat{z}(s_c, a_{c,j'}). \tag{46}$$

Further, each penalty term $\hat{z}(s_k, a)$ is non-negative, because the terms are defined as $\hat{z}(s_k, a) = v^{\mathbb{F}}_{k+1}(s_k, a) - v^{\mathbb{I}}_{k+1}(s_k, a)$ and

$$v^{\mathbb{I}}_{k+1}(s_k, a) = \mathbb{E}\left[\min_{\pi \in \Pi_{\rho^{\text{TSP}}}} \sum_{k'=k}^{K} C(s_{k'}, X^{\pi}_{k'}(s'_k))\right] \leq \min_{\pi \in \Pi_{\rho^{\text{TSP}}}} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X^{\pi}_{k'}(s'_k))\right] = v^{\mathbb{F}}_{k+1}(s_k, a).$$

The reversal of expectation and minimization that produces the middle inequality is a result of the use of perfect information in the construction of $v^{\mathbb{I}}_{k+1}(s_k, a)$. As a result, $\hat{z}(s_c, a_{c,j'}) \leq \hat{z}(s_j, a_{j,c}) + \hat{z}(s_c, a_{c,j'})$, so if we can show that

$$\hat{z}(s_j, a_{j,j'}) \leq \hat{z}(s_c, a_{c,j'}), \tag{47}$$

then we are done.

Writing the penalties explicitly and somewhat abusing notation for epoch indices, inequality (47) is equivalent to

$$v_{j+1}^{\mathbb{F}}(s_j, a_{j,j'}) - v_{j+1}^{\mathbb{I}}(s_j, a_{j,j'}) \leq v_{c+1}^{\mathbb{F}}(s_c, a_{c,j'}) - v_{c+1}^{\mathbb{I}}(s_c, a_{c,j'}). \tag{48}$$

Notice, however, that each term represents an expected cost-to-go from node j'. The terms on the left-hand side represent costs-to-go from node j' after traveling directly from j, while terms on the right-hand side represent costs-to-go after first balking CS c. Notice also that, for a given filtration, the cost-to-go from node j' cannot be better after balking at CS c than if having traveled directly. To prove this is the case, we refer the reader to Lemma 1.

Thus, $v_{j+1}^{\mathbb{F}}(s_j, a_{j,j'}) \leq v_{c+1}^{\mathbb{F}}(s_c, a_{c,j'})$ and $v_{j+1}^{\mathbb{I}}(s_j, a_{j,j'}) \leq v_{c+1}^{\mathbb{I}}(s_c, a_{c,j'})$, so equation (48) holds, meaning the triangle inequality also does in the presence of penalties. □

By Proposition 4, because the optimal policy for the penalized perfect information problem is AC, we can write its objective function as

$$\begin{aligned}
\mathbb{E}\left[\min_{\pi\in\Pi}\sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) + \hat{z}\,(s_k, X_k^\pi(s_k))\right] &= \mathbb{E}\left[\min_{\pi\in\Pi^{\text{AC}}}\sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) + \hat{z}\,(s_k, X_k^\pi(s_k))\right] \\
&= \mathbb{E}\left[\min_{\rho\in\mathcal{R}(s_0)}\left\{\min_{\pi\in\Pi_\rho}\sum_{k=0}^{K} C\,(s_k, X_k^\pi(s_k)) + \hat{z}\,(s_k, X_k^\pi(s_k))\right\}\right].
\end{aligned}$$

Restricting our search to the set of AC policies is especially convenient, because there are significantly fewer charging decisions to consider in the inner minimization.

Lemma 1 (Unimproved cost-to-go after balking a CS)**.** *Consider a vehicle in some state s_j at location j. The cost-to-go from a location j' as measured by the TSP static estimating policy is no greater if the vehicle travels directly from j to j' than if it travels j to $c \in \mathcal{C}$, balks c, then travels c to j'.*

Proof. Proof. Denote by $s_{k(j')}$ the resulting state of the vehicle that traveled directly j to j', and $s_{k(cj')}$ the resulting state of the vehicle that first balked at CS c. Recall that the TSP Static policy performs a single iteration of the outer minimization of equation (15), then solves the FRVCP for the resulting CL sequence. The CL sequence ρ^{TSP} resulting from a single solution of the master problem (16)-(22) will be the same for both $s_{k(j')}$ and $s_{k(cj')}$, so what we must show is that the value of the optimal policy produced by the solution to the subproblem for this sequence is no worse from state $s_{k(j')}$:

$$\min_{\pi\in\Pi_{\rho^{\text{TSP}}}} \mathbb{E}\left[\sum_{k=k(j')}^{K} C(s_k, X_k^\pi(s_k))\right] \leq \min_{\pi\in\Pi_{\rho^{\text{TSP}}}} \mathbb{E}\left[\sum_{k=k(cj')}^{K} C(s_k, X_k^\pi(s_k))\right]. \tag{49}$$

The left-hand side of (49) corresponds to the objective when traveling directly, and the right-hand side corresponds to the objective after balking. We proceed by contradiction.

For the statement (49) to be false, it must be the case that there is an action available *downstream* from state $s_{k(cj')}$ (in epochs $\{k(cj'), \ldots, K\}$) that yields a lower objective value and is not available downstream from state $s_{k(j')}$. As described in §4.3.2, the subproblem consists in finding the optimal charging decisions along ρ^{TSP} and can be modeled as a dynamic program with action space defined by (38)-(41). By the definition of this action space, the only actions exclusively available downstream from state $s_{k(cj')}$ are those in equation (39) that correspond to charging decisions to energy levels less than that with which the vehicle would arrive downstream from state $s_{k(j')}$. For such charging decisions to be in the set of feasible actions, it must be that the charge level is sufficient to reach the next stop in the CL sequence $n^\star$ and some subsequent CS c'. However, if this were the case, then – by the triangle inequality – the vehicle downstream from state $s_{k(j')}$ could simply skip the CS visit and instead proceed directly to $n^\star$, which would result in less incurred cost. Thus, it is not the case that there exists an action downstream from state $s_{k(cj')}$ that yields a lower objective value and is not available from state $s_{k(j')}$, so (49) holds. □ □

G Proof of Proposition 3

We begin by repeating the statement for Proposition 3:

Let $\mathcal{A}_{CV}(s_k)$ be a relaxation of action space $\mathcal{A}(s_k)$ defined by the removal of conditions $(\exists c \in \mathcal{C} : a^q \geq e_{a^i c})$ *in equation* (4) *and* $(q_k \geq e_{i_k a^i})$ *in equation* (5). *Further, let Π_{CV} be the set of feasible policies under $\mathcal{A}_{CV}$. Then there exists an optimal policy $\pi^\star \in \Pi_{CV}$ that does not visit any charging stations.*

Proof. Proof. First, we note that the feasibility of $\pi^\star$ is guaranteed by the construction of $\mathcal{A}_{\text{CV}}$, since the relaxed conditions ensure that the vehicle can always relocate to an unvisited customer or a CS.

We proceed by contradiction. If $\pi^\star$ is not optimal, then there exists a policy π that does visit CSs and has a lower objective value. However, it is easy to construct a policy π' with better performance by following policy π, except when it chooses to visit CSs. In those cases, π' advances directly to the next customer visited by π (or the depot, if terminating). In so doing, the objective value of π' will be no greater than that of π. But this contradicts our assumption that π has a strictly lower objective value, so it must be that an optimal policy exists that does not visit any CSs. □ □

H Disaggregated Results of Computational Experiments

Table 5 contains disaggregated results for the computational experiments described in §7. The naming convention for the instances (or, more accurately, the technician assignments) is "*geography_zoneID-technicianID*." Optimal static, PreOpt, and Private-only entries marked with asterisks denote instances that we were able to solve to optimality; the rest are the best solutions found after three hours of computation. Empty cells for the PI bound denote instances for which we could not solve at least 38/50 realizations of uncertainty to optimality (all PI values shown are optimal). Entries marked "inf" are infeasible under the private-only recharging strategy.

Table 5: Disaggregated objective values over testbed of instances, including CV bound and the objective value under the private-only recharging strategy. *Note: Values in minutes. "inf" entries (Private-only) were infeasible. We set a solver time limit of 3 hrs. Throughout, asterisks (*) indicate optimality. For PI, this implies optimal solutions to all 50 realizations of uncertainty for the instance. For non-optimal solutions, Optimal Static, PreOpt, and QPreOpt use the solver's best feasible solution; PI uses the solver's best bound.*

			low								moderate								high							
Instance	CV	Private-only	PI	QPostTSP	QPreOpt	PreOpt	Opt Static	PostTSP	TSP Static	Myopic	PI	QPostTSP	QPreOpt	PreOpt	Opt Static	PostTSP	TSP Static	Myopic	PI	QPostTSP	QPreOpt	PreOpt	Opt Static	PostTSP	TSP Static	Myopic
rural_18-1	180.06	inf	200.62*	201.30	204.48*	204.60*	206.04*	204.04	207.19	335.87	201.74	207.68	209.88	209.75	213.22	208.98	214.91	362.73	203.34	243.32	255.09	271.27	269.96	291.04	294.64	514.24
rural_18-3	176.89	inf	198.09*	205.28	203.80*	203.56*	203.47*	203.66	204.32	267.83	203.52	228.03	231.87	229.45	224.76*	231.00	228.81	306.58	207.15	319.77	343.82	371.79	367.18	453.07	356.86	478.37
rural_18-4	174.45	inf	195.14*	197.60	197.60*	197.60*	197.60*	197.60	198.50	363.29	207.49	218.09	218.09*	218.09*	218.09*	218.09	219.00	382.45	223.27	310.46	310.46*	310.46	310.46	310.46	311.11	469.50
rural_18-5	107.25	172.64	110.21*	111.10	111.79*	111.80*	112.53*	112.77	113.47	196.17	111.63	116.71	119.37	119.23	121.84	120.31	125.66	217.17	113.21	138.50	148.19	152.42	165.10	153.94	167.98	307.26
rural_18-6	118.24	238.77	123.83	125.95	129.04	129.04	129.04	129.04	128.13	219.31	123.85	130.25	132.37	132.37	132.37	132.37	135.98	221.57	123.99	153.55	176.83	176.83	176.83	176.83	177.11	268.02
rural_19-0	211.91	inf	239.98	246.89	247.27	247.27	248.55	247.27	247.44	338.70	240.18	271.65	275.10	274.28	283.43	274.41	284.18	382.98	240.72	384.56	409.08	419.35	419.09	484.09	424.31	564.90
rural_19-1	170.13	inf	187.96*	193.11	192.98	192.37	193.26	192.37	193.26	296.02	188.46	200.98	200.10	200.10	200.10	203.91	221.47	314.12	188.57	234.95	245.08	245.08	245.08	247.84	260.41	397.98
rural_19-3	126.72	inf	134.71	136.69	138.42	137.19	137.34	137.19	138.70	246.13	134.75	159.13	168.14	161.48	157.83	161.48	157.33	264.86	134.70	220.74	218.86	218.86	218.86	225.14	243.03	349.48
rural_20-0	155.46	inf	171.14*	174.76	174.76	174.76	174.76*	174.76	178.23	288.19	171.89	195.53	195.53	195.53	195.53	195.53	197.93	306.83	172.03	249.51	286.04	286.04	286.04	286.04	293.23	389.35
rural_20-3	111.07	133.54*	115.48*	116.52	118.40*	118.40*	118.40*	118.40	117.65	175.14	117.68*	122.47	127.15*	130.78*	129.86*	130.78	128.56	170.89	126.28*	131.02	133.54*	133.54*	133.54*	133.54	134.04	170.73
rural_20-4	111.74	219.61	115.89*	116.87	117.75	117.97*	119.77*	117.97	119.77	176.55	116.40	134.82	139.11	138.25	138.14	138.25	138.14	199.73	117.06	178.99	219.62	219.62	219.62	219.62	220.98	288.27
rural_20-5	211.65	inf	241.54	247.30	249.45	248.77	248.43*	248.83	249.45	302.58	241.79	285.53	286.89	286.20	284.10	296.19	286.72	339.85	242.16	450.81	470.43	465.96	463.79	475.35	469.06	523.24
rural_20-6	194.83	inf	219.47*	229.33	229.52	229.14	229.14*	229.14	229.14	361.75	220.80	254.51	256.13	253.63	253.49	253.51	253.08	396.66	222.32	360.17	385.76	389.12	388.72	452.32	388.71	561.61
rural_20-7	113.56	inf	118.58*	122.05	122.05*	121.64*	121.83*	121.64	122.07	161.06	120.39	138.38	142.88*	141.96*	142.06*	141.96	142.03	182.19	124.22	188.48	235.15	234.31	233.90	234.31	234.14	273.99
rural_21-0	135.33	149.12*	145.16*	147.51	146.41	146.41	147.41*	146.79	148.07	271.02	145.86*	150.26	149.12*	149.12*	149.12*	150.33	150.35	278.63	147.73*	150.47	149.12*	149.12*	149.12*	150.33	150.35	395.36
rural_21-1	125.24	306.32	132.71*	133.26	133.92	133.78	134.09	133.78	134.95	205.06	132.83	137.22	135.85	138.94	139.41	138.94	139.21	239.18	132.90	168.87	160.72	170.49	181.47	170.49	182.86	420.12
rural_21-3	132.46	inf	141.34*	143.14	142.01	142.59	143.62	142.59	144.95	216.27	141.41	145.38	143.71	146.31	146.95	146.31	150.80	235.44	141.50	171.63	167.37	179.54	191.14	179.54	192.20	324.35
rural_21-4	214.31	inf	243.56	253.71	252.32	251.34	250.60	250.97	250.60	301.56	243.69	279.97	286.46	290.34	288.93	289.54	288.93	339.91	244.76	349.43	436.99	451.62	450.90	450.77	450.94	516.10
rural_21-6	203.13	inf	229.21*	233.72	234.52*	233.94*	234.54*	233.82	233.95	279.82	231.55*	243.24	244.80	246.77	245.8*	244.21	244.69	304.47	238.64	294.28	298.14	320.71	327.13	392.39	330.00	430.18
rural_21-7	145.68	277.89	157.66*	160.03	159.94	160.07*	160.31*	160.07	160.31	239.55	158.85	185.50	185.50	183.86	180.27*	183.19	180.27	259.11	162.39	239.02	278.77	277.85	275.04	277.00	275.04	349.12
rural_21-8	111.96	155.82*	116.55*	120.94	120.94*	119.39*	119.11*	119.39	119.11	192.77	123.36	139.19	145.42	140.63	139.61*	140.63	139.61	192.77	125.98	154.46	155.82*	155.82*	155.82*	155.82	159.93	192.77
rural_22-0	196.02	inf	219.99	227.79	226.48	224.82	225.48	224.81	225.48	265.70	219.98	249.39	257.61	249.31	249.31	249.31	249.31	303.61	219.97	297.86	302.44	340.65	348.95	330.96	349.69	479.57
rural_22-1	220.08	inf	250.53*	257.40	256.79	259.14	258.86	258.12	257.14	318.58	251.87	274.89	281.54	280.47	283.43	280.21	282.76	372.99	251.95	345.04	371.34	395.22	411.48	394.95	421.85	570.96
rural_22-2	135.15	inf	144.49*	146.34	146.37	145.26	148.03	145.26	147.81	199.11	144.58	154.38	161.69	161.69	161.69	167.79	167.50	220.42	144.96	189.30	207.92	207.92	207.92	210.20	215.59	308.18
rural_22-3	125.28	inf	132.59*	135.00	134.97	134.94	135.43	134.94	135.43	194.47	132.65	151.37	152.24	154.17	155.12	154.17	155.12	213.91	132.73	194.35	226.74	226.74	226.74	226.74	229.42	302.55
rural_22-4	105.36	163.03	108.21*	110.41	112.32*	112.32*	112.32*	112.32	110.07	262.88	108.96*	111.63	116.31*	116.31*	116.31*	116.31	127.12	344.01	110.42	130.20	160.37	160.37	160.37	160.37	167.94	653.08
rural_22-5	98.04	192.49*	98.99*	100.19	100.43*	100.43*	101.21*	100.43	101.21	139.21	101.01*	104.77	110.48*	111.53*	112.99*	111.53	111.31	142.32	115.74*	130.95	138.82	143.36*	153.71*	143.36	154.97	214.66
semi_urbain_18-8	120.87	262.96	151.42	159.02	156.41	156.41	176.69	158.45	179.72	274.88	151.01	177.23	193.90	194.75	193.74	196.09	199.85	359.88	151.72	212.66	235.59	247.13	243.79	246.74	247.94	674.10
semi_urbain_20-1	100.21	207.76	132.74	164.97	158.24*	158.24*	158.24*	171.95	166.28	237.36	133.29	195.84	207.85	207.85	207.85	208.50	248.70	321.40	133.22	203.46	207.85	207.85	207.85	208.50	266.75	631.11
semi_urbain_21-7	104.75	inf	128.46	129.93	130.17*	130.78*	131.47*	130.78	129.29	189.43	128.87	141.15	140.77	141.23	141.23	141.23	142.96	241.66	130.36	179.16	178.08	189.32	189.32	189.32	186.97	376.19
semi_urbain_21-11	111.95	inf	127.24	134.77	134.82	135.29	154.13	135.29	156.16	240.34	127.08	157.07	185.72	187.81	184.50	187.81	186.93	317.61	127.08	209.91	223.42	241.27	232.78	241.27	235.02	627.97
semi_urbain_22-0	97.24	200.04*	131.02*	154.98	151.89	162.93*	148.43*	162.93	151.35	203.78	151.61*	191.15	200.04*	200.04*	200.04*	200.04	232.48	283.80	179.10*	197.37	200.04*	200.04*	200.04*	200.04	275.92	592.54
semi_urbain_22-1	131.65	inf	156.14	173.58	183.29	188.89	181.70*	188.89	183.87	244.79	156.45	213.87	263.39	274.22	256.39	274.22	259.85	324.64	156.72	291.84	314.71	329.15	325.08	329.66	339.43	628.56
semi_urbain_22-2	110.34	128.83*	128.83*	128.83	128.83*	128.83*	128.83*	128.83	132.41	173.79	128.83*	128.83	128.83*	128.83*	128.83*	128.83	132.41	234.71	128.83*	128.83	128.83*	128.83*	128.83*	128.83	132.41	374.80

I Tabular Summary of Notation

We provide in Table 6 a list of most notation used throughout the manuscript.

Table 6: Notation used in the manuscript, presented in order of appearance. *Note: *These symbols may also appear with a superscript a, referring to their value in the post-decision state after taking action a in epoch k*

Notation	Description
$\mathcal{N}$	Set of customers
N	Number of customers
$\mathcal{C}$	Set of CSs
$\mathcal{C}'$	Set of extradepot CSs
V	Set of all nodes ($\mathcal{C} \cup \mathcal{N}$)
$t_{i,j}$, $e_{i,j}$	Time and required to travel between node i and node j
ψ_c	Number of charging terminals at CS c
$p_{c,\text{arrive}}, p_{c,\text{depart}}$	Probability of arrival to and departure from CS c per time
Q	EV's battery capacity
$\mathcal{Q}$	Set of discrete energy levels to which the EV may choose to recharge
K	Number of epochs (terminal epoch)
s_k	*State of the system at the beginning of epoch k
t_k	*Time of the system at the beginning of epoch k
i_k	*Location of the vehicle at the beginning of epoch k
q_k	*Energy in the vehicle's battery at the beginning of epoch k
$\bar{\mathcal{N}}_k$	*Unvisited customers at the beginning of epoch k
z_k	*Vehicle's position in queue at the beginning of epoch k
$\mathcal{M}_k$	*Vehicle's memory of queue observations at the beginning of epoch k
$\mathcal{S}$	State space
$\mathcal{A}(s_k)$	Action space for state s_k
$a = (a^i, a^q)$	Action a, a (location, charge) pair
$\mathcal{I}(s_k^a)$	Information space for post-decision state s_k^a
$W_{k+1} = (w^t, w^q)$	Exogenous information observed in epoch k, a (time, position in queue) pair
$\bar{u}(e_1, e_2)$	Function specifying the time to charge from energy e_1 to e_2
$C(s_k, a)$	Cost of taking action a from state s_k
Π	Set of all Markovian deterministic policies
$X_k^\pi(s_k)$	Decision rule for policy π in epoch k mapping state s_k to an action
$\tau(\pi)$	Objective value of policy π
$p_j = (p_j^i, p_j^q)$	jth direction of fixed route p, a (location, charge) pair
$\pi(p)$	Fixed-route policy derived from fixed route p
P	Set of all feasible fixed routes
Π^{S}	Set of all feasible fixed-route policies
$R(\pi(p))$	Sequence of locations visited under fixed-route policy $\pi(p)$
$r(\pi(p))$	CL sequence for fixed-route policy $\pi(p)$
$\mathcal{R}(s_k)$	Set of all possible CL sequences from state s_k
$\text{Sym}(\bar{\mathcal{N}}_k)$	Set of all permutations of set $\bar{\mathcal{N}}_k$
Π^{AC}	Set of all feasible AC policies
Π^{B}	Set of all feasible fixed-route policies that visit a CS without recharging
$T_D(\rho)$	Direct-travel cost of CL sequence ρ
$Y^\star(\rho)$	Optimal charging decisions for CL sequence ρ
$T(\rho, Y^\star(\rho))$	Cost of CL sequence ρ given charging decisions $Y^\star(\rho)$
$e_j^\star$	Maximum energy with which an EV can arrive to location j
$\mathcal{P}_\rho$	Set of substrings of sequence ρ of length 2+
$\bar{\mathcal{P}}_\rho$	Elements of $\bar{\mathcal{P}}_\rho$ not traversible without recharging
Π_ρ	Set of all fixed-route policies with CL sequence ρ
$z_j = (z_j^t, z_j^q)$	jth supporting point in FRVCP labeling algorithm, a (time, charge) pair
$\mathcal{S}_{\text{post}}(s_k)$	Reachable post-decision states from state s_k
$\pi^{\text{TSP}}(s_k)$	TSP static policy from state s_k
Π_{CV}	Set of all policies that are not required to visit a CS
$\mathbb{F}$	Natural filtration
$\mathbb{I}$	Perfect information filtration
$\mathscr{F}_k$	Information known under filtration $\mathbb{F}$ at the beginning of epoch k