Dynamic Electric Vehicle Routing: Heuristics and Dual Bounds

Nicholas D Kullman, Justin Goodson, Jorge E. Mendoza

Dynamic Electric Vehicle Routing:
Heuristics and Dual Bounds

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, we show the value of our policies to be within 4.7 percent of the value of an optimal policy in most instances. Further, we demonstrate that our policies significantly outperform the industry-standard routing strategy in which vehicle recharging generally occurs at a central depot. More broadly, we offer examples for how operations research tools classically employed in static and deterministic routing can be adapted for dynamic and stochastic routing problems.

1 Introduction

Electric vehicles (EVs) are replacing 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 (2018) are increasingly incorporating EVs into their commercial fleets, which have historically been comprised of CVs. EVs are also being adopted as the go-to transport method 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 central 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*. In private-only operations, companies effectively limit their use of EVs to urban environments in which they can cover their routes on a single charge. With this restriction, meeting customer demand may require more EVs than CVs, leading to higher capital costs and thereby reducing the competitiveness of EVs relative to CVs.

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 offering companies 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. 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, impeding their broader adoption and, ultimately, the transition to sustainable transportation. In this work, we aim to alleviate these problems by providing routing solutions that specifically address the uncertainty in using public charging infrastructure and thereby allow for use of the public-private recharging strategy.

Our work draws on a broad spectrum of operations research tools, many of which are commonly employed in static and deterministic routing. Importantly, our research serves as a template for how these tools can be adapted for dynamic and stochastic routing problems. For instance, we show how fixed routes, a staple in deterministic routing, can be used in the construction of good policies and in the establishment of dual bounds via information relaxations and nonlinear information penalties. While the use of fixed routes in stochastic and dynamic routing problems itself is not novel (e.g., Novoa and Storer (2008); Goodson et al. (2013); Goodson et al. (2016)), we formalize the concept via a decomposition that allows for broader application of these tools. Illustrating this connection – between static, deterministic routing and dynamic, stochastic routing – we claim as a broad contribution. More specifically, we make the following contributions:

- 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 (Powell 2011), which allows the route planner to adapt to realized demand at public CSs.

- We offer a decomposition of the E-VRP-PP that is valid for a subclass of policies. The decomposition allows the use of machinery from static and deterministic routing in solution methods for our stochastic and dynamic routing problem.

- We propose static and dynamic routing policies, including the optimal static policy. To construct static policies, we employ a Benders-based branch-and-cut algorithm to solve the decomposition of the E-VRP-PP. We then incorporate these static policies into dynamic lookaheads (rollouts), in

which they serve as base policies. We implement three lookaheads of varying depths: pre-decision (zero-step), post-decision (half-step), and one-step rollouts.

- Using the same decomposition and Benders-based algorithm in conjunction with an information relaxation, we establish the value of the optimal policy with perfect information, which serves as a dual bound on the value of an optimal non-anticipative 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. (2018) to solve this new variant exactly.

- We construct and apply nonlinear information penalties that tighten the perfect information bound on the optimal policy. A first in vehicle routing, this serves as a proof-of-concept and provides a template for future applications of information penalties in this field. Our penalties reduce by half the gap between the perfect information dual bound and our best routing policy. To the best of our knowledge, this is also the first successful application of information penalties to a combinatorial perfect information problem lacking any special structure that makes the problem more tractable. We demonstrate how to construct penalties using tools familiar to the vehicle routing community, including TSP-like math programs, decompositions, and fixed routes.

- Using the established bounds, we demonstrate that our routing policies are competitive with the optimal policy, within 4.7% for most instances. Given that even the existence of a dual bound is uncommon in the stochastic and dynamic routing literature, this is a notable result.

- We show that all of our policies under the public-private recharging strategy outperform the optimal solution under the private-only recharging strategy, with our best policies offering savings of over 20% on average. These results lend further motivation for companies to adopt the public-private recharging strategy and significantly extend EVs' utility in commercial applications.

- In general, our work offers an example of how to connect static and deterministic routing methods with dynamic and stochastic routing problems.

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 discuss 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 $\mathcal{N} = \{1, \ldots, N\}$ and CSs $\mathcal{C} = \{0, N+1, \ldots, N+C\}$ 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 increasing, is also still an exception to the norm.)

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 ($p_{c,x}$ and $p_{c,y}$, respectively): $\mathrm{M/M/}\psi_c/\infty$. 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 selectable 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 Problem Model

We formulate the E-VRP-PP as an MDP whose components are defined as follows.

2.2.1 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 $s_k = \left(t_k, i_k, i_{k-1}, q_k, q_{k-1}, \bar{\mathcal{N}}_k, z_k\right)$, 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 and its location in the previous epoch $i_k, i_{k-1} \in V$; the energy currently in the vehicle's battery and at the beginning of the previous epoch $q_k, q_{k-1} \in [0, Q]$; the set of customers that have not yet been visited $\bar{\mathcal{N}}_k \subseteq \mathcal{N}$; and the vehicle's position in queue at its current location $z_k \in \mathbb{N}$. We define $z_k = 1$ when $i_k \in \{0\} \cup \mathcal{N}$. This definition of the system state yields the state space $\mathcal{S} = \mathbb{R}_{\geq 0} \times V \times V \times [0, Q] \times [0, Q] \times \mathcal{N} \times \mathbb{N}$. The system is initialized in epoch 0 at time 0 with the vehicle at the depot, the battery at maximum capacity,
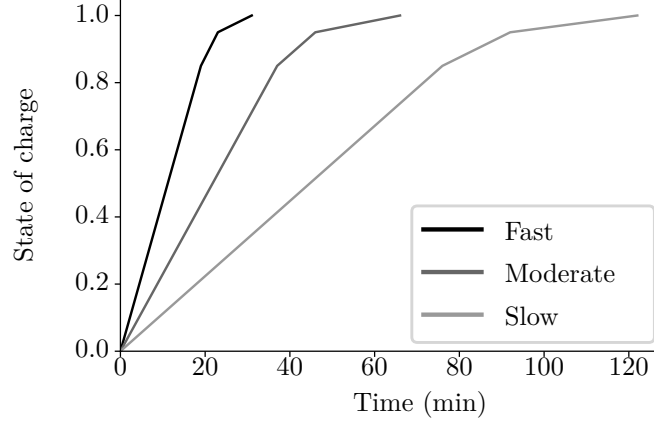
Figure 1: The vehicle's charging function for different charging technologies. We assume a concave piecewise-linear charging function as in Montoya et al. (2017).

and all customers yet to be visited:

$$s_0 = (0, 0, 0, Q, 0, \mathcal{N}, 1). \tag{1}$$

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

2.2.2 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

$$
\begin{aligned}
\mathcal{A}(s_k) = \Big\{ & (a^i, a^q) \in \{\bar{\mathcal{N}}_k \cup \mathcal{C}\} \times [0, Q] : \\
& a^i = i_k, a^q = q_k, \\
& \quad i_k \in \mathcal{C}' \wedge \psi_{i_k} < z_k \\
& a^i = i_k, a^q \in \Big\{ \tilde{q} \in \mathcal{Q} \Big| \tilde{q} > q_k \wedge \\
& \big((\exists c \in \mathcal{C}, \exists j \in \bar{\mathcal{N}}_k : \tilde{q} \geq e_{i_k j} + e_{jc}) \vee (\bar{\mathcal{N}}_k = \emptyset \wedge \tilde{q} \geq e_{i_k 0}) \big) \Big\}, \\
& \quad i_k \in \mathcal{C} \wedge z_k \leq \psi_{i_k} \wedge q_k \leq q_{k-1} \\
& a^i \in \bar{\mathcal{N}}_k, a^q = q_k - e_{i_k a^i}, \\
& \quad (\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) \wedge (i_k \neq i_{k-1} \vee q_k \neq q_{k-1}) \\
& a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i}, \\
& \quad (i_k \neq i_{k-1} \vee q_k \neq q_{k-1}) \wedge q_k \geq e_{i_k a^i} \\
& \quad \wedge (q_k > q_{k-1} \Rightarrow (k = 0 \vee (\bar{\mathcal{N}}_k = \emptyset \wedge a^i = 0))) \Big\}.
\end{aligned}
$$

$$\tag{2}$$
$$\tag{3}$$
$$\tag{4}$$
$$\tag{5}$$

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 and did not charge in the previous epoch. 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. In addition, we require that the vehicle must not have queued in the previous epoch, because we require that an EV always charge after queuing. Finally, equation (5) defines routing decisions to charging station nodes. Again, we require that the vehicle not have queued in the previous epoch and that it have sufficient charge to reach the charging station. We also disallow visits to CSs after the vehicle has just charged except when the EV is initially departing the depot and when it has served all customers and is en route to terminate at the depot.

2.2.3 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 $s_k^a = \left(t_k^a, i_k^a, i_{k-1}^a, q_k^a, q_{k-1}^a, \bar{\mathcal{N}}_k^a, z_k^a \right)$. In s_k^a we have updated the vehicle's previous location and charge to the location and charge in epoch k: $i_{k-1}^a = i_k$, $q_{k-1}^a = q_k$. The vehicle's new current location and charge are inherited from the action: $i_k^a = a^i$, $q_k^a = a^q$. Finally, we update the set of unvisited customers: $\bar{\mathcal{N}}_k^a = \bar{\mathcal{N}}_k \setminus \{a^i\}$. The time and position in queue remain unchanged from the pre-decision state.

2.2.4 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 position in queue and the time, which were unchanged in the pre-to-post-decision transition. In the first two epoch-triggering events, our transition to s_{k+1} may be stochastic and depend on the observation of exogenous information. For instance, if in the first case we arrive at an extradepot CS, then we observe exogenous information in the form of the queue length. In the second case, when we have waited at an extradepot CS, 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

$$\mathcal{I}(s_k^a) = \Bigg\{ (w^t, w^z) \in \mathbb{R}_{\geq 0} \times \mathbb{N} :$$

$$w^t = t_k^a + t_{i_{k-1}^a i_k^a}, w^z = 1,$$
$$i_k^a \in \mathcal{N} \cup \{0\} \wedge i_k^a \neq i_{k-1}^a \tag{6}$$

$$w^t = t_k^a + \bar{u}(q_{k-1}^a, q_k^a), w^z = z_k^a,$$
$$q_k^a > q_{k-1}^a \tag{7}$$

$$w^t = t_k^a + t_{i_{k-1}^a i_k^a}, w^z \in \mathbb{N},$$
$$i_k^a \neq i_{k-1}^a \wedge i_k^a \in \mathcal{C}' \tag{8}$$

$$w^t \in (t_k^a, \infty), w^z = \psi_{i_k^a},$$
$$i_k^a = i_{k-1}^a \wedge q_k^a = q_{k-1}^a \Bigg\}, \tag{9}$$

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. The probabilities of observing a given queue length or waiting time in equations (8) and (9), respectively, follow from well known $M/M/\psi_c/\infty$ queue dynamics.

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 $z_{k+1} = w^z$. The other state components, all of which were updated in the transition to the post-decision state, remain the same.

2.2.5   Contribution Function.

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

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

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.

2.2.6 Objective Function.

Let Π denote the set of Markovian deterministic policies, where a policy $\pi \in \Pi$ is a sequence of decision rules $(X_0^\pi, X_1^\pi, \ldots, X_K^\pi)$ where each $X_k^\pi : s_k \to \mathcal{A}(s_k)$ is a function mapping the current state s_k to an action in the action space $\mathcal{A}(s_k)$. 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 which case a policy is defined as a set of decision rules from epoch k onwards: $(X_k^\pi, X_{k+1}^\pi, \ldots, X_K^\pi)$.

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 problems 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 coverage of E-VRP research, a common shortcoming in nearly all existing works is the lack of consideration of access to public charging infrastructure. 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 the EVs to begin charing 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 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 consider utilizing 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 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 research in stochastic and dynamic routing for EVs, especially in the domain of E-VRPs, inhibiting the adoption of the public-private recharging strategy.

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: $p = (p_1, p_2, \ldots, p_{|p|})$, where each direction $p_j = (p_j^i, p_j^q)$ is a location-charge pair. Let us consider a vehicle in the state $s_1 = (t_{0,3}, 3, 0, Q - e_{0,3}, Q, \{1, 2\}, 1)$, as in
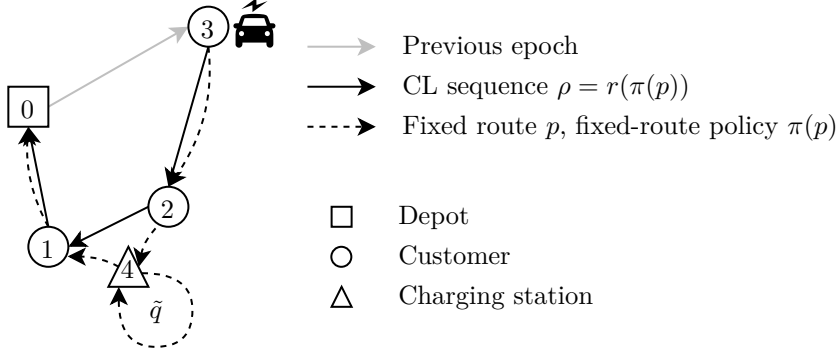
Figure 2: Shown is an EV that relocated from the depot to customer 3 in epoch 0. The CL sequence ρ (solid black arrows) considered by the vehicle from its current state is $(3, 2, 1, 0)$. The fixed route p (dashed black arrows) includes a detour to CS 4 where it charges to $\tilde{q}$, as indicated by the self-directed arc at 4 (p is given by equation (12)).

Figure 2. We might consider the fixed route

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

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

Fixed-Route Policies. The sequence of directions comprising a fixed route p constitutes a *fixed-route policy*, equivalently, a *static policy* $\pi(p) \in \Pi$, which is defined by decision rules

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

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 $\left(i_k = p_{j^\star - 1}^i \wedge q_k = p_{j^\star - 1}^q \wedge \bar{\mathcal{N}}_k = \left(\bigcup_{l=j^\star}^{|p|} p_l^i\right) \setminus \mathcal{C}'\right)$. 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 2 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 remove uncertainty from the problem (see §6.1), 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 $\Pi^S \subseteq \Pi$, defined as the set $\Pi^S = \{\pi(p) \in \Pi \,|\, p \in P\}$ 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 are unchanging in their decision making given a fixed route p. Aside from choosing to wait when required, all decisions are made in advance, and the policies provide no meaningful way in which to respond to uncertainty.

CL Sequences. Given a fixed-route policy $\pi(p) \in \Pi^S$, let us denote by $R(\pi(p))$ the sequence of locations visited: $R(\pi(p)) = (p_j^i)_{j \in \{1,\ldots,|p|\}}$. In the above example, $R(\pi(p)) = (3,2,4,4,1,0)$. Notice that some of the locations in the sequence $R(\pi(p))$ 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 ($p_1 = 3$), its ending point (the depot, $p_{|p|} = 0$), 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,

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

We call sequences like $r(\pi(p))$ *compulsory-location* (CL) *sequences*. The longer sequences $R(\pi(p))$ 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 sequences like $R(\pi(p))$ as being the energy-feasible analogs to CL sequences.

Our current definition of CL sequences (14) is based on a fixed route p. Often, however, we wish to find CL sequences based on the vehicle's current state rather than a predetermined 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, 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* $\Pi^{AC} \subseteq \Pi^S$ (AC for "always charge"). The set of AC policies is defined as $\Pi^{AC} = \Pi^S \setminus \Pi^B$, where $\Pi^B = \left\{\pi(p) \in \Pi^S \,\middle|\, p \in P \wedge \left(\exists j \in \{2,\ldots,|p|-1\} : p_j^i \in \mathcal{C} \wedge p_{j-1}^i \neq p_j^i \wedge p_j^i \neq p_{j+1}^i\right)\right\}$ is the set of static policies that include CS visits at which no charging is performed. Note that the sets Π^{AC} and Π^B are mutually exclusive and collectively exhaust the set of static policies, meaning $\Pi^{AC} \cap \Pi^B = \emptyset$ and $\Pi^{AC} \cup \Pi^B = \Pi^S$. We justify the restriction of static policies to AC policies in the proof of Theorem 1.

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

Proof. See §B.

Theorem 1 provides a valuable result, as the restriction to AC policies significantly reduces the set of fixed routes and fixed-route policies that we must consider. Going forward, all mention of static or fixed-route policies, unless explicitly stated otherwise, refers 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 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 given the constraints established by the action space (2)-(5). We offer the following decomposition:

Theorem 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. 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 and in the design of nonlinear information penalties.

We note that the decomposition (15) also provides a bridge between our dynamic and stochastic routing problem and the static and deterministic routing literature, allowing solution methods for fixed routes developed in the latter to be used for the former. Decomposing a dynamic and stochastic routing problem so as to permit the use of static and deterministic routing tools is likely not unique to the E-VRP-PP. Many dynamic routing problems where fixed-route policies are applicable may be amenable to a similar decomposition.

4.3 Solving the Decomposed E-VRP-PP

To solve (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.

Before proceeding, we highlight an additional benefit of the decomposition – namely, beyond serving as a gateway between static and dynamic routing, the result of Theorem 2 also permits the use of the Benders-based solution method. This allows us to solve the decomposed problem (right-hand side of (15)) exactly for instances of nontrivial size. In contrast, our experience solving exactly the non-decomposed problem (left-hand side of (15)) using a large mixed-integer linear program (MIP) suggests it is computationally intractable for all but the simplest instances.

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 $\mathcal{M}_k = i_k \cup \bar{\mathcal{N}}_k \cup \{0\}$. The master problem

approximates the cost of 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. If indeed charging operations are required for the vehicle to traverse ρ, this is accounted for in the objective of the subproblem.

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 $\mathcal{M}_k \subset V$. This yields the following master problem:

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

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

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

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

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset \mathcal{M}_k, |S| \geq 2 \tag{20}$$

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

If $i_k \neq 0$, to ensure the CL sequence starts at i_k and ends at the depot, we add the following constraint:

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

where the corresponding arc cost is zero ($t_{0i_k} = 0$). 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 the 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 $\{j | x_{\rho_{i-1} j} \in \mathbf{x}_\rho\}$. The objective function (16) contains the variable θ whose value depends on the Benders cuts produced in the subproblem.

4.3.2 Subproblem: Charging.

The master problem (16)-(22) produces a CL sequence ρ. The master problem's approximation of the cost of ρ considers only direct-travel costs, ignoring any costs associated with charging operations that may be required to ensure energy feasibility. The subproblem's task is to remedy this, determining the optimal charging decisions for the given sequence ρ and adjusting the objective value (16) accordingly through the variable θ. 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 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 §6.1. 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
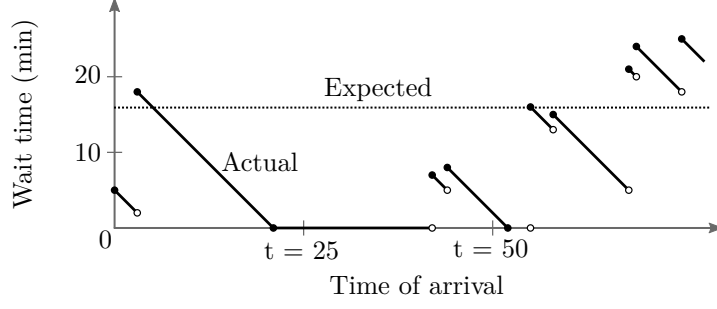
Figure 3: Waiting times at an extradepot CS under natural and perfect information filtrations. Under the natural filtration, the operator only has access to the expected waiting time at the CS (dashed line), whereas under the perfect information filtration, they are aware of the actual waiting time (solid line), which depends on time of arrival.

expected waiting time at c. Alternatively, if we assume the decision maker is operating under the perfect information 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. For a depiction of waiting times under natural and perfect information filtrations, see Figure 3.

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 follow 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:

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

$$
a^i = i_k, a^q = q_k,
$$
$$
i_k \in \mathcal{C}' \wedge \psi_{i_k} < z_k \tag{23}
$$
$$
a^i = i_k, a^q \in \{\tilde{q} \in \mathcal{Q} | \tilde{q} > q_k \wedge (\exists c \in \mathcal{C} : \tilde{q} \geq e_{i_k n^\star} + e_{n^\star c})\},
$$
$$
i_k \in \mathcal{C} \wedge z_k \leq \psi_{i_k} \wedge q_k \leq q_{k-1} \tag{24}
$$
$$
a^i = n^\star, a^q = q_k - e_{i_k a^i},
$$
$$
(\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) \wedge (i_k \neq i_{k-1} \vee q_k \neq q_{k-1}) \wedge (i_k \in C \Rightarrow q_k > q_{k-1}) \tag{25}
$$
$$
a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i},
$$
$$
(i_k \neq i_{k-1} \vee q_k \neq q_{k-1}) \wedge q_k \geq e_{i_k a^i} \wedge (i_k \in C \Rightarrow q_k > q_{k-1})
$$
$$
\wedge (q_k > q_{k-1} \Rightarrow (k = 0 \vee (\bar{\mathcal{N}}_k = \emptyset \wedge a^i = 0))) \Big\}. \tag{26}
$$

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 (25) and (26). 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 (25). 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 (24). Under the definition in equation (24),

14

when the vehicle charges, it must do so to an energy level sufficient to reach the next location $n^\star$.

To solve the subproblem DP, we use the exact labeling algorithm for the FRVCP proposed by Froger et al. (2018). However, the FRVCP under consideration here requires discrete charging decisions and, for the FRVCP-P, the inclusion of time-dependent waiting times. We modify the labeling algorithm to account for these two additional features. The algorithm and our modifications to it are discussed in more detail in §4.4.

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 θ_ρ is $T(\rho, Y^\star(\rho)) - T_D(\rho)$. We add Benders optimality cuts to the master problem so the objective function value for ρ reflects the total fixed-route cost $T(\rho, Y^\star(\rho))$ rather than only the direct travel costs $T_D(\rho)$:

$$\theta \geq \theta_\rho \left(\left(\sum_{x \in \mathbf{x}_\rho} x \right) - (|\mathbf{x}_\rho| - 1) \right) \tag{27}$$

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 (27) is at most 0, which is redundant given the non-negativity constraint on θ (21). Note that because the triangle inequality holds across travel times and energy consumption, and waiting and charging times are non-negative, $\theta_\rho \geq 0$ always.

The optimality cuts (27) 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 the *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: $\bar{\mathcal{P}}_\rho = \left\{ \sigma \in \mathcal{P}_\rho \big| e^\star_{\sigma_1} < e_{\sigma_1 \sigma_2} + e_{\sigma_2 \sigma_3} + \cdots + e_{\sigma_{|\sigma|-1} \sigma_{|\sigma|}} \right\}$, 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 (such as for the FRVCP-P), 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 CL sequence ρ 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: $\sum_{x \in \mathbf{x}_\rho} x \le |\mathbf{x}_\rho| - 1$.

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 $\sum_{x \in \mathbf{x}_{\rho'}} x \le |\mathbf{x}_{\rho'}| - 1$, 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. (2018) 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 electronic companion, §D. 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 4). Labels are defined by *state-of-charge* (SoC) *functions*. (To maintain consistency with Froger et al. (2018), 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 5 for an 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 5, which depicts this process when extending a label along the edge from node 0 to node $4a$ in Figure 4. 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
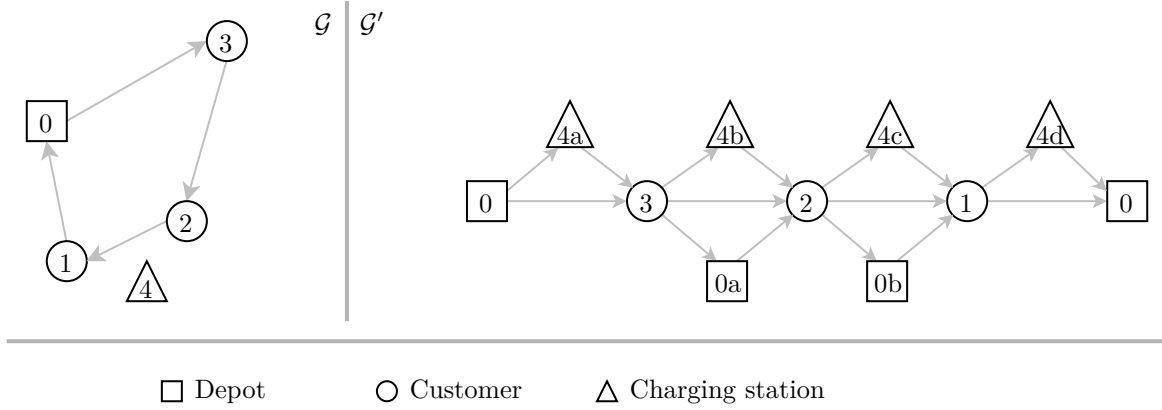
Figure 4: Left is an example of an original graph $\mathcal{G}$ for the E-VRP-PP. The gray path in the figure shows a CL sequence ρ. Right shows the corresponding modified graph $\mathcal{G}'$ used to model and solve the FRVCP, which includes a dummy node for each possible CS visit.

diamond in the left graph of Figure 5. We then consider the set of possible charging decisions at that CS. The right graph of Figure 5 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. (2018).

With our modifications (§D), 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.

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.

5.1 Static Policies

The decomposition in Theorem 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
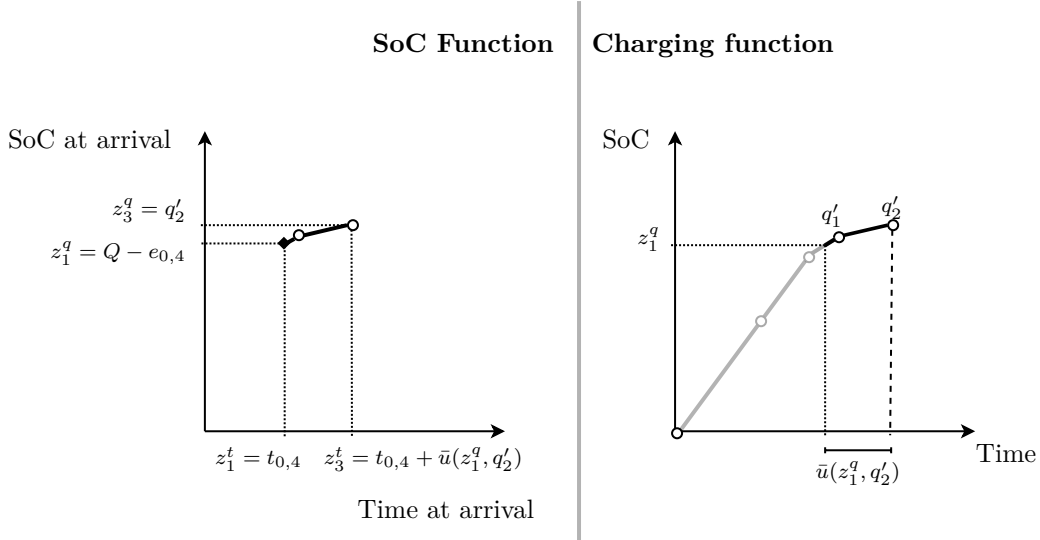
Figure 5: Depicting the creation of new supporting points at CS nodes for the case of node 4a in Figure 4. Left shows the SoC function at node 4a. The initial supporting point is the black diamond $(z_1 = (t_{0,4}, Q - e_{0,4}))$. We create additional supporting points (z_2 and z_3, circles) for each 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.

computationally expensive, we also consider an approximation which we call the TSP Static policy. For both, following from Theorem 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 $\Pi^{\text{AC}} \subseteq \Pi$. 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 (1), the shortest Hamiltonian path corresponds to the optimal TSP tour over $\mathcal{N}'$.) We then optimally solve the FRVCP for ρ^{TSP} to generate an energy-feasible fixed route whose corresponding fixed-route policy is 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 three dynamic policies for the E-VRP-PP, all of which leverage rollout algorithms.

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 (Goodson et al. 2017). 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, a post-decision rollout with a TSP Static base policy, and a one-step rollout with a myopic 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 solved iteratively 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 in the E-VRP-PP 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 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 $\bar{\mathcal{N}}_k^a \cup \{0\}$ to produce a CL sequence, then solve an FRVCP on that sequence to produce the base policy $\pi^{\mathrm{TSP}}(s_k^a)$. The post-decision rollout decision rule is then to select an action $a^\star$ solving

$$\min_{a \in \mathcal{A}(s_k)} \left\{ C(s_k, a) + \mathbb{E}\left[\sum_{i=k+1}^K C(s_i, X_i^{\pi^{\mathrm{TSP}}(s_k^a)}(s_i)) \middle| s_k \right] \right\}. \tag{28}$$

5.2.3 One-step Rollout of Myopic Policy.

Consider again action selection from a state s_k in which the vehicle just visited a customer. The pre-decision rollout executes a single base policy from the pre-decision state s_k, and the post-decision rollout executes $\bar{N} + C$ base policies, one from each reachable post-decision state $s_k^a \in \mathcal{S}_{\text{post}}(s_k)$. One-step rollouts consider all reachable pre-decision states in the next epoch $k + 1$. When we consider relocating to a customer node or to the depot, there is only one reachable pre-decision state (because the post-to-pre-decision transition is deterministic). However, for relocations to extradepot CSs $c \in \mathcal{C}'$, if there is a nonzero probability of observing a queue up to some length L_c at CS c, then there are $L_c + 1$ reachable pre-decision states. Thus, with the one-step rollout at epoch k, we must execute a total of $\bar{N} + 1 + \sum_{c \in \mathcal{C}'}(L_c + 1)$ base policies. This prohibits the use of even the TSP Static base policy. Instead, we use a myopic base policy π^{m}. We refer to the one-step rollout of the myopic policy as *OSM*.

Denote by $\mathcal{S}(s_k, a)$ the set of all reachable pre-decision states in epoch $k + 1$ given s_k and an action $a \in \mathcal{A}(s_k)$. From each $s_{k+1} \in \mathcal{S}(s_k, a)$, we execute the myopic base policy $\pi^{\text{m}}(s_{k+1})$, which at each epoch simply chooses actions with minimum immediate cost. Taking the expectation over the set of all reachable pre-decision states in $\mathcal{S}(s_k, a)$ provides an estimate for the cost-to-go associated with action a. Formally, the decision rule for OSM is to select an action $a^\star$ solving

$$\min_{a \in \mathcal{A}(s_k)} \left\{ C(s_k, a) + \sum_{s_{k+1} \in \mathcal{S}(s_k, a)} \mathbb{E}\left[\sum_{i=k+1}^{K} C(s_i, X_i^{\pi^{\text{m}}(s_{k+1})}(s_i)) \middle| s_{k+1} \right] \cdot \mathbb{P}\left(s_{k+1} | s_k, a\right) \right\}. \quad (29)$$

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 establish a dual bound using the expected value of an optimal policy with perfect information, i.e., the performance achieved via a clairvoyant decision maker. To establish this bound, we solve the decomposed E-VRP-PP under a perfect information relaxation. In §6.2 we then 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. 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 E-VRP-PP.

6.1 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

$$\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{k=0}^K C(s_k, X_k^\pi(s_k)) \middle| s_0\right] = \mathbb{E}\left[\min_{\pi \in \Pi}\sum_{k=0}^K C(s_k, X_k^\pi(s_k)) \middle| s_0\right]. \tag{30}$$

Notice that the *perfect information problem* (30) 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 solid line in Figure 3, 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 (30), 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 Theorem 1, which applies to static policies regardless of information filtration, we may restrict our search to AC policies. Further, as demonstrated in Theorem 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

$$\mathbb{E}\left[\min_{\pi(p) \in \Pi^{AC}} \sum_{k=0}^K C(s_k, X_k^{\pi(p)}(s_k)) \middle| s_0\right] = \mathbb{E}\left[\min_{\rho \in \mathcal{R}(s_0)}\left\{\min_{\pi \in \Pi_\rho}\sum_{k=0}^K C\left(s_k, X_k^\pi(s_k)\right)\right\} \middle| s_0\right]. \tag{31}$$

To solve the inner 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 Information Penalties

In reality, because no decision maker is clairvoyant and advanced knowledge of the future is often valuable, the dual bound achieved with perfect information may be loose. 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

$$\mathbb{E}\left[\min_{\pi \in \Pi} \sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) + z\left(s_k, X_k^\pi(s_k)\right)\Bigg| s_0\right]. \tag{32}$$

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 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 (32) 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 $v_{k+1}^\mathbb{G}(s_k, a)$ 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 $\hat{z}(s_k, a) = v_{k+1}^\mathbb{F}(s_k, a) - v_{k+1}^\mathbb{I}(s_k, a)$. To compute $v_{k+1}^\mathbb{G}(s_k, a)$ 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 $\mathbb{G}$: $v_{k+1}^\mathbb{G}(s_k, a) = \mathbb{E}\left[\sum_{i=k+1}^{K} C\left(s_i, X_i^{\pi(s_{k+1}, \mathbb{G})}(s_i)\right)\Big| s_k, a\right]$. For our estimating policy, we use the TSP static policy (see §5.1.2). Then we may write our penalty explicitly as

$$\hat{z}(s_k, a) = \mathbb{E}\left[\sum_{i=k+1}^{K} C\left(s_i, X_i^{\pi^{\text{TSP}}(s_{k+1}, \mathbb{F})}(s_i)\right)\Bigg| s_k, a\right] - \mathbb{E}\left[\sum_{i=k+1}^{K} C\left(s_i, X_i^{\pi^{\text{TSP}}(s_{k+1}, \mathbb{I})}(s_i)\right)\Bigg| s_k, a\right] \tag{33}$$

The objective for the penalized PI problem with our approximation is

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

$$= \mathbb{E}\left[\min_{\pi \in \Pi} \sum_{k=0}^{K} C\left(s_k, X_k^\pi(s_k)\right) + v_{k+1}^\mathbb{F}\left(s_k, X_k^\pi(s_k)\right) - v_{k+1}^\mathbb{I}\left(s_k, X_k^\pi(s_k)\right)\Bigg| s_0\right]. \tag{34}$$

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 Theorem 3.

Theorem 3. *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* (32), *where the penalty is $\hat{z}$ as defined in equation* (33). *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. See §E.

Following from Theorems 2 and 3, we may decompose the penalized perfect information problem into routing and charging decisions as before, so the objective function becomes

$$\mathbb{E}\left[\min_{\rho \in \mathcal{R}(s_0)}\left\{\min_{\pi \in \Pi_\rho} \sum_{k=0}^{K} C\left(s_k, X_k^\pi(s_k)\right) + v_{k+1}^\mathbb{F}\left(s_k, X_k^\pi(s_k)\right) - v_{k+1}^\mathbb{I}\left(s_k, X_k^\pi(s_k)\right)\right\}\Bigg| s_0\right]. \tag{35}$$

We can again estimate the objective value of (35) using simulation, as we did to estimate the unpenalized objective value with perfect information in (31). The inner minimization of (35) is still an

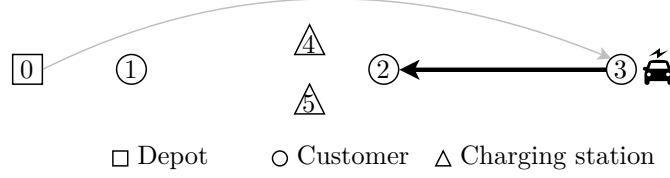□ Depot   ○ Customer   △ Charging station

Figure 6: 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.

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 Theorem 3 is crucial, as it significantly reduces the number of realizable states that must be enumerated in the reaching algorithm. 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 6 with the vehicle in state $s_1 = (t_{0,3}, 3, 0, Q - e_{0,3}, Q, \{2, 1\}, 1)$. We assume CSs 4 and 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 $s_2 = (t_{0,3} + t_{3,2}, 2, 3, Q - e_{0,3} - e_{3,2}, Q - e_{0,3}, \{1\}, 1)$. 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

23

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. The exercise is not in vain, however, since methods that yield near-optimal policies for smaller instances may portend toward good methods for larger instances. In addition, the demonstration of nonlinear information penalties is a first in vehicle routing and serves as a proof-of-concept.

7 Computational Experiments

We perform experiments that assess the quality of our routing policies relative to one another, relative to the perfect information bound, and relative to the dual bound established using information penalties. However, as instance size increases, we are restricted in which experiments we can perform. We begin in §7.2 by discussing experiments on the smallest instances, for which we can compute both the perfect information dual bound and the dual bound with information penalties. In these experiments, we compare our best static and dynamic policies against these bounds. Next, in §7.3 we describe experiments for larger, 10-customer instances. At this size, we can no longer solve for the dual bound with information penalties, but we can still compute and compare our policies against the perfect information dual bound. Then in §7.4 we describe experiments on our largest set of instances, which have between 20 and 40 customers. For instances of this size, we can no longer establish a dual bound, nor compute the optimal static policy. As a result, we simply compare the performance of the TSP static, PostTSP and OSM policies. Finally, in §7.5 we revisit the question of recharging strategies. We compare the optimal solution under the private-only strategy to the performance of our policies operating under the public-private recharging strategy. Prior to discussing any experimental results, we first describe in §7.1 how we generated our instances.

7.1 Instance Generation

In creating our instances, we use many of the same parameters as in the case study by Villegas et al. (2018), because it is largely this study and its results on recharging strategies that motivate this work. While the battery capacity used is somewhat dated, this does not affect the relevance of our methods nor the impact of our results, since it is not the battery capacity itself but rather the relationship between range and route length that is important.

The depot, customers, and charging stations lie within a 100 km × 100 km square area. The vehicle has a battery capacity of $Q = 16$ kWh, travels at a speed of 40 km/hr, and consumes energy at a rate of 7.5 kW/km. We assume all chargers at a CS have the same technology, which may be either *fast*,

medium, or *slow*; the depot is assumed to be equipped with fast charging technology. The charging functions for each technology are those given in Montoya et al. (2017), which have breakpoints (changes in charge rate) at $0.85Q$ and $0.95Q$. These values are always members of the set of chargeable energy levels $\mathcal{Q}$; additional energy levels in $\mathcal{Q}$ vary by instance and are discussed in later subsections. The number of chargers at each extradepot charging station c is ψ_c and also varies by instance. For a CS c with fast, medium, or slow charging technology, the mean service time μ_c is 20, 45, or 80 minutes, respectively. The departure rate for a charger at extradepot CS c is equal to $p_{c,y} = 1/\mu_c$. The utilization at extradepot charging stations u_c, i.e., the probability of all chargers being occupied upon a vehicle's arrival, again varies by instance. Together with the number of chargers ψ_c and the departure rate $p_{c,y}$, this fixes the inter-arrival rate at c according to $p_{c,x} = u_c \cdot p_{c,y} \cdot \psi_c$. We assume infinite system capacity so that a vehicle will never be stranded – they can always choose to wait. In practice, however, we use a finite value for system capacity ℓ_c, chosen such that it is practically infinite. That is, that the limiting probability of observing more than $\ell_c + 1$ vehicles in the queue is less than some $0 < \epsilon \ll 1$. Making queues finite allows us to limit the number of states considered when performing rollouts. All instances for our computational experiments are available on the Vehicle Routing Problem Repository (Mendoza et al. 2014).

7.2 Experiments with Information Penalties

To demonstrate the utility of information penalties we seek instances for which access to perfect information is exceptionally valuable, producing 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 construct such an instance 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 (35)) 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. In addition, we limit the set of chargeable battery states $\mathcal{Q}$ to the charging function breakpoints and multiples of 25% ($\mathcal{Q} = \{0, 0.25Q, 0.5Q, 0.75Q, 0.85Q, 0.95Q, Q\}$).

The experimental results for this instance over 250 samples of uncertainty are shown in Figure 7. 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 non-anticipative 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. These results also showcase the advantages offered by dynamic decision making, as our best dynamic policy, PostTSP, outperforms the optimal static policy by almost 30% relative to the PI bound.

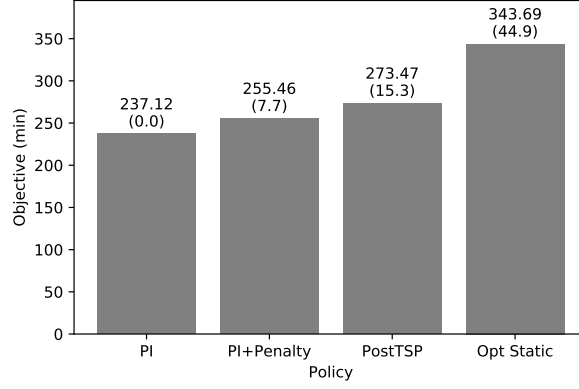Computation times for these experiments are shown in the first column of Table 1. We see that,

Figure 7: Comparing our best non-anticipative dynamic (PostTSP) and static ("Opt Static") policies to the dual bounds afforded by the value of the optimal policy with perfect information ("PI") and the value of the optimal policy with penalized access to perfect information ("PI + Penalty"). Bar labels are average objective achievement over 250 samples of uncertainty with percent difference from the PI bound in parentheses.

Table 1: Computation times (in ms) for experiments by instance size (number of customers).

	4 (Penalty)	10	20	30	40
PI+Penalty[†]	5.9E+05	-	-	-	-
PI[†]	22	9.9E+05	-	-	-
Opt Static[‡]	140	1.0E+06	-	-	-
PreOpt[*]	-	526	-	-	-
TSP Static[‡]	-	75	85	210	358
OSM[*]	-	1	9	28	97
PostTSP[*]	1	38	109	507	1.8E+03

∗ *Values represent average time to make a decision in each epoch*

† *Values represent average time to solve each realization of uncertainty*

‡ *Values represent average time to solve each instance*

per realization of uncertainty, it takes on average almost 10 minutes to find the optimal solution with information penalties. The average time per realization to find the perfect information solution is 22 ms. More time (140 ms) was required to find the optimal static solution to the penalty instance, although this only has to be done once (rather than per realization of uncertainty). For our single reported dynamic policy, PostTSP required on average less than 0.5 ms per epoch to select an action.

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.

7.3 Policy Performance for 10-Customer Instances

Next, we consider a set of one hundred 10-customer instances. To produce this set of instances, we create 20 random geographies of customers and charging stations across the 100 km × 100 km area, then produce five different CS-demand scenarios for each. Of the 20 geographies, 10 have 2 extradepot CSs, and 10 have 4 extradepot CSs. All extradepot CSs $c \in \mathcal{C}'$ are assumed to have three chargers ($\psi_c = 3$), and their technology is randomly assigned according to unif{"fast", "medium", "slow"}. To produce the CS-demand scenarios, we randomly draw utilizations for each extradepot CS: $u_c \sim$ unif{.45, .60, .75, .90}. The set of chargeable battery states $\mathcal{Q}$ consists of the charge function breakpoints and multiples of 10%.

The results of the experiments for the 10-customer instances over 50 samples of uncertainty are shown in Figure 8 (disaggregated results are also available in the electronic companion, §F). We were able to solve the optimal static policy for 99/100 instances and were able to solve for the perfect information bound for 4973 out of the 5000 experiments (100 instances × 50 samples of uncertainty). Note that there is no penalized dual bound for this set of instances. It required an average of almost 10 minutes to solve for a single realization of uncertainty for the four-customer instance, and with computations increasing exponentially with the number of customers, it is simply intractable for larger instances given our current methods.

There is effectively a tie for the best performing policy between the optimal static policy and PreOpt, both of which are within 4.7% of the optimal policy as determined by the dual bound provided by the PI relaxation. It is interesting to note that, in contrast to the penalty instance, a static solution method performs at least as well as our best dynamic solution method. In constructing the penalty instance, it was designed such that knowledge of exogenous information was particularly valuable. Unable to respond to this information as it was realized, static solution methods were unable to compete with the dynamic decision making methods. In these 10-customer instances, no such emphasis was placed on making exogenous information particularly valuable. Thus, we see similar performance between dynamic and static solution methods. In general, the proximity of PreOpt and the optimal static policy to the dual bound is a strong result, and it demonstrates the value of an optimal static solution.

The PostTSP policy is within 6.0% of the dual bound, which is 2.1% better than its static counterpart, the TSP Static policy. The objective value of the TSP Static policy trails the optimal static policy by roughly 3.4%, with a gap of about 8.1% to the perfect information bound. This difference reflects the value of finding the optimal CL sequence in the outer minimization of equation (15). Finally, we see that all fixed-route-based policies outperform the one-step rollout of the myopic policy. The latter achieves an objective value roughly 22% higher than the perfect information bound.

Computation times for these experiments are listed in the second column of Table 1. On average, per realization of uncertainty, it takes over 16.5 min to find the solution with perfect information. This is similar to the average time to compute the optimal static policy (16.6 min), although the latter need be computed only once. Compared to the 16.6 min to generate the optimal static policy, the TSP Static policy can be generated in about 75 ms – nearly five orders of magnitude faster. For our dynamic policies, the PreOpt policy takes on average about 0.5 s to make a decision in each epoch; however, this time assumes that the optimal static policy has already been identified in the initial state and is available in epoch zero. The PostTSP and OSM policies are fastest, requiring only 37.7 ms and 1.4 ms, respectively, to select an action in each epoch.

This set of 10-customer instances is the largest for which we can demonstrate our policies' performance relative to a dual bound. While 10 customers is not a particularly large instance size, it is a reasonable assumption of the number of customers a single vehicle may visit in a day. For larger, multi-vehicle
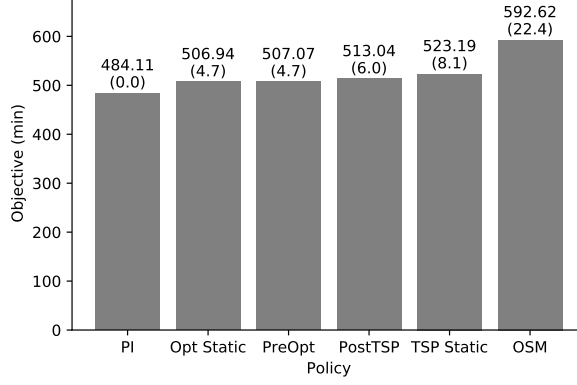
Figure 8: Performance of policies for the 10-customer instances. Labels are average objective achievement over 50 samples of uncertainty with percent difference from the PI bound in parentheses.

settings, it is reasonable to think that routes or customer groups would be assigned to vehicles ahead of time. From there, our proposed solution methods could be used for day-of execution, and bounds could be established.

7.4 Policy Performance for 20-40 Customer Instances

Finally, we consider three sets of larger instances: 20 customers with 3 extradepot CSs, 30 customers with 4 extradepot CSs, and 40 customers with 5 extradepot CSs. Each set consists of 25 instances, comprised of five random geographies of the nodes and five CS-demand scenarios for each geography. As for the 10-customer instances, we assume extradepot CSs $c \in \mathcal{C}'$ have three chargers ($\psi_c = 3$) and that the charging technology is randomly assigned according to unif{"fast", "medium", "slow"}. The CS-demand scenarios are created by randomly drawing utilizations for each extradepot CS, as before: $u_c \sim \text{unif}\{.45, .60, .75, .90\}$. The chargeable battery states $\mathcal{Q}$ again consist of the charge function breakpoints and multiples of 10%.

Figure 9 provides results for the 20-40 customer instances (disaggregated results are again available in the electronic companion, §F). Here, no dual bound is available against which to compare the value of our policies, as the size of the instances prohibits an exact solution to equation (15). For the same reason, the optimal static policy and PreOpt are also absent from this analysis. We implement the PostTSP, OSM, and TSP static policies, and compare their performance relative to the best policy – PostTSP. PostTSP outperforms the TSP Static policy by 5.8% and the OSM policy by 16.7%. This gap between PostTSP and the TSP Static policy is nearly three times larger than it was for the set of 10-customer instances, suggesting that the value of dynamic decision making may increase with problem complexity.

The computation times for these experiments are listed in the third through fifth columns of Table 1. The TSP Static policy can still be generated quickly, requiring only 358 ms for the 40-customer instances, 210 ms for the 30-customer instances, and 85 ms for the 20-customer instances. The best performing dynamic policy, PostTSP requires non-negligible time to conduct its rollouts, especially for the 40-customer instances in which it takes more than 1800 ms on average to select an action. For 30-customer instances, PostTSP requires just over 500 ms to select an action, and just over 100 ms for the 20-customer instances. Relative to the PostTSP policy, OSM sacrifices performance for speed, averaging less than 97 ms to select an action for the 40-customer instances, less than 28 ms for the 30-customer instances, and
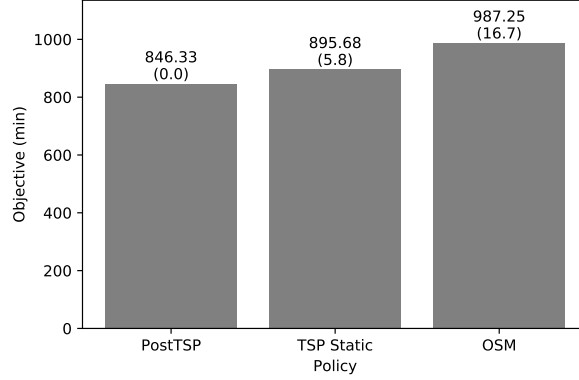
28

Figure 9: Performance of policies for the 20-40 customer instances. Labels are average objective achievement over 50 samples of uncertainty with percent difference from the best policy (PostTSP) in parentheses.

less than 10 ms for the 20-customer instances.

7.5 Public-Private vs. Private-Only Recharging Strategies

Motivating our research on the E-VRP-PP is companies' current aversion to uncertainty that has so far led to their use of a private-only recharging strategy. This is despite evidence (Villegas et al. 2018) suggesting that a public-private recharging strategy offers additional flexibility and potential cost savings. We wish to provide further evidence in favor of the public-private recharging strategy, demonstrating that even in the face of uncertainty at public charging stations, our policies perform favorably relative to the private-only strategy. We provide this evidence in Figure 10, which shows for the 10-customer instances the performance of our policies relative to the optimal private-only solution. To obtain the optimal private-only solutions, we remove extradepot CSs from the instances and solve the perfect information problems. We find that all proposed policies, even the One-Step Myopic policy, outperform the optimal private-only solution. Our best-performing policies outperform the optimal private-only solution by more than 20%. This suggests that by committing to a private-only recharging strategy, companies may be losing out on significant savings, hampering EVs' adoption in commercial applications.

Yet, as EVs continue to increase in popularity and related technologies develop, it is likely that this gap will widen further. Assuming charging infrastructure increases at a rate similar to its demand, the average performance of policies should improve. Consider, for instance, Figure 11, which compares policies' performance under 2 and 4 extradepot CSs in the 10-customer instances. Indeed, all policies' performance improved, both in raw objective units and relative to the PI bound. 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. In fact, access to this real-time information may be modeled as a relaxation for the current problem. That is, we could 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 it $\mathbb{Q}$). The performance of the decision maker under $\mathbb{Q}$ would be bounded below by the dual bounds and above by our current best-performing policy, since $\mathbb{F} \subseteq \mathbb{Q} \subseteq \mathbb{I}$.
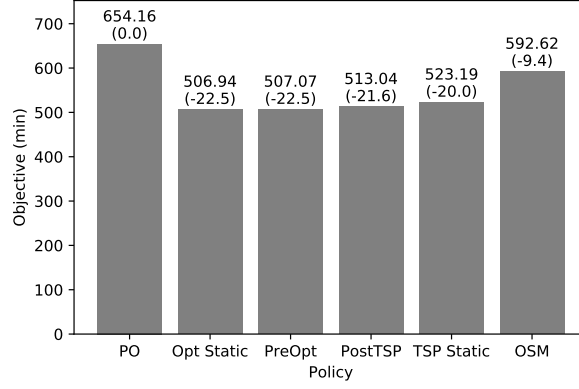
Figure 10: Performance of policies for the 10 customer instances relative to the optimal private-only (PO) solution. All policies, including OSM, outperform the optimal private-only solution.
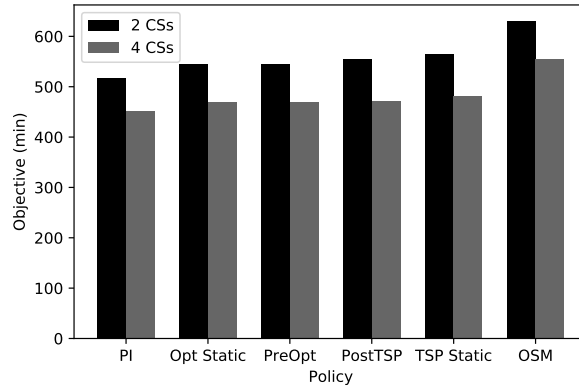

Figure 11: Demonstrating the difference in performance of policies between 10 customer instances with 2 extradepot CSs and 10 customer instances with 4 extradepot CSs.

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. We also establish a dual bound on the value of the optimal policy through the use of a perfect information relaxation. Using that dual bound, we have demonstrated in computational experiments that our routing policies are competitive with the optimal policy, coming within 4.7% for most instances. To tighten the dual bound, we successfully imposed nonlinear information penalties, a first in vehicle routing.

Our work was grounded in 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 provide evidence that with a good dynamic routing policy, such companies could enact a public-private recharging strategy and that doing so would be cheaper than the private-only strategy. We showed this to be true, demonstrating that all of our policies under the public-private recharging strategy soundly outperform the optimal solution under a private-only strategy, with our best policies offering savings of more than 20%. 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.

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.

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`, Accessed on July 24, 2018.

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.

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 July 24, 2018.

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 and Operations Research*, 2018. Accepted with minor revisions.

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

J. Goodson, J. W. Ohlmann, and B. Thomas. Rollout policies for dynamic solutions to the multi-vehicle routing problem with stochastic demand and duration limits. *Operations Research*, 61(1):138–154, 2013.

J. Goodson, B. Thomas, and J. Ohlmann. Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. *Transportation Science*, 50(2):363–761, 2016.

Justin C. Goodson, Barrett W. Thomas, and Jeffrey W. Ohlmann. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research*, 258 (1):216 – 229, 2017. ISSN 0377-2217.

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.

La Poste. Instantané : Les véhicules électriques de La Poste, 2018. https://legroupe.laposte.fr/decouverte/les-vehicules-electriques-de-la-poste, Accessed on July 24, 2018.

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.

C. Novoa and R. Storer. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196(2):509–515, 2008.

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 22, 2018.

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.

Warren B Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, volume 842. John Wiley & Sons, 2011.

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 September 22, 2018.

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 July 24, 2018.

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`.

Accompanying Material

Content in the electronic companion is organized as follows. We first provide a definition for the set of fixed routes in §A. Then we provide a proof of Theorem 1 in §B and Theorem 2 in §C. In §D we describe our modifications made to the algorithm of Froger et al. (2018) to solve the FRVCP exactly in consideration of discrete charging decisions and time-dependent waiting times. We then offer the proof of Theorem 3 in §E, and finally present a disaggregated view of the results to our computational experiments in §F.

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:

$$\mathcal{A}^-(s_k) = \left\{ (a^i, a^q) \in \{\bar{\mathcal{N}}_k \cup \mathcal{C}\} \times [0, Q] : \right.$$

$$a^i = i_k, a^q \in \left\{ \tilde{q} \in \mathcal{Q} \middle| \tilde{q} > q_k \wedge \right.$$

$$\left. \left((\exists c \in \mathcal{C}, \exists j \in \bar{\mathcal{N}}_k : \tilde{q} \geq e_{i_k j} + e_{jc}) \vee (\bar{\mathcal{N}}_k = \emptyset \wedge \exists c \in \mathcal{C} : \tilde{q} \geq e_{i_k c} \wedge e_{c0} \leq Q) \right) \right\},$$

$$i_k \in \mathcal{C} \wedge q_k \leq q_{k-1} \tag{36}$$

$$a^i \in \bar{\mathcal{N}}_k, a^q = q_k - e_{i_k a^i},$$

$$(\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) \tag{37}$$

$$a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i},$$

$$q_k \geq e_{i_k a^i}$$

$$\left. \wedge (q_k > q_{k-1} \Rightarrow (k = 0 \vee (\bar{\mathcal{N}}_k = \emptyset \wedge a^i = 0))) \right\}. \tag{38}$$

In contrast to the definition of $\mathcal{A}(s_k)$, there is no condition on position in queue in order to charge in (36). Further, we remove waiting actions from the action space, so we also remove the conditions on not relocating if having just waited in equations (37) and (38). 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

$$P_1 = \{(i_{k'}, q_{k'})\}$$

$$P_2 = \mathcal{A}^-(s_{k'})$$

$$\vdots$$

$$P_j = \bigcup_{s' \in \mathcal{S}^-(s_{(k'+j-3)}, p_{j-1})} \mathcal{A}^-(s')$$

$$\vdots$$

$$P_D = \{(0, q) | q \in [0, Q]\}$$

B   Proof of Theorem 1

Theorem. *Good fixed-route policies are AC Policies. 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. In order for a policy $\pi \in \Pi^B$ 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 $\tau(\pi^{AC}) \leq \tau(\pi)$. 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. $\qquad\square$

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. This construction strategy for π^{AC} 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 Theorem 2

Theorem. *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 $\pi(p) \in \Pi^{AC}$ maps to a CL sequence $r(\pi(p))$ given by equation (14), we may equivalently write the set of AC policies as $\Pi^{AC} = \bigcup_{\rho \in \mathcal{R}(s_k)} \Pi_\rho$, where $\Pi_\rho = \{\pi(p) \in \Pi^{AC} : r(\pi(p)) = \rho\}$. 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:

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

$\qquad\square$

D   Modifications to Froger et al. (2018) algorithm for the FRVCP

Froger et al. (2018) 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
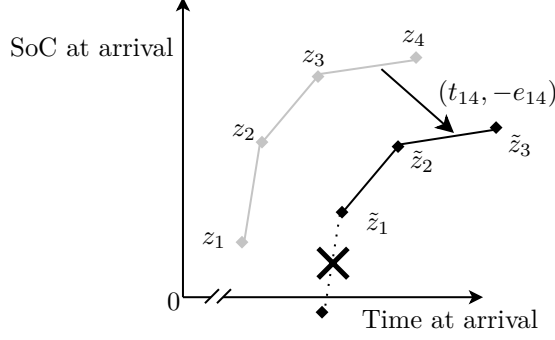
Figure 12: An example of shifting the SoC function as we extend the label along the edge from customer node 1 to CS node 4d in Figure 4. The SoC function for node 1 (in gray) is translated by $(t_{14}, -e_{14})$. The resulting SoC function for the label at node 4d (in black) contains one fewer supporting point, since the translation of z_1 yields an infeasible point with negative SoC.

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. (2018), 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 4. 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 12.

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 can enter service at the CS. To do so, after the initial translation (depicted in Figure 12), we shift the SoC function supporting points again according to the underlying wait time. Define the function $w : (\mathbb{R}_{\geq 0} \times \mathcal{C}) \to \mathbb{R}_{\geq 0}$ that specifies how long the EV must wait if it arrives to some CS c at some time t. Because functions $w(t, c')$ are not generally continuous for a given CS c', 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 4. 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 12. Now, in Figure 13 we consider 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 $w(\tilde{z}_j^t, 4)$ to produce the final SoC functions for these labels.

Figure 13 depicts an example for the FRVCP-P, in which waiting times are time-dependent. For the
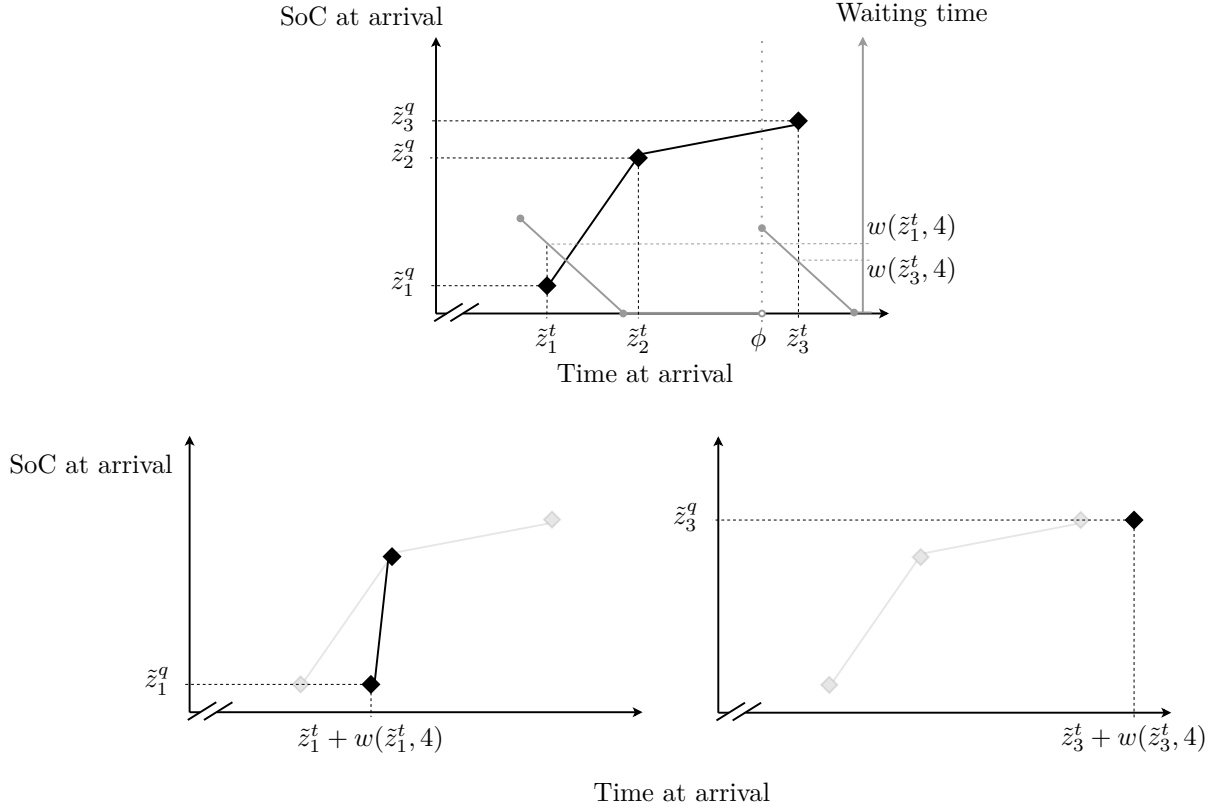
Figure 13: Depiction of handling time-dependent waiting times. In the top graph, we have the resulting SoC function after the initial translation from node 1 to node 4d depicted in Figure 12. This is superimposed over the wait-time function $w(t, 4)$, plotted in gray. The supporting points for the SoC 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.

FRVCP-N, waiting times are constant, so there are no discontinuities in $w(t, c')$, and there is no need to divide the supporting points and create multiple labels. Instead, all supporting points are simply shifted by the constant waiting time value. We note that for both time-dependent and constant waiting times, labels' resulting supporting points are still guaranteed to produce a concave SoC function, because the queues obey the first-come-first-served property.

E Proof of Theorem 3

Theorem. *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 (32), where the penalty is $\hat{z}$ as defined in equation (33). 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. We proceed similarly as in the proof of Theorem 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 $\tau_{\hat{z}}(\pi^{AC}) \leq \tau_{\hat{z}}(\pi)$. 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 Theorem 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

$$\hat{z}(s_j, a_{j,j'}) \leq \hat{z}(s_j, a_{j,c}) + \hat{z}(s_c, a_{c,j'}). \tag{39}$$

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

$$v_{k+1}^{\mathbb{I}}(s_k, a) = \mathbb{E}\left[\min_{\pi \in \Pi_{\rho^{\mathrm{TSP}}}} \sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi}(s_k')) \right] \leq \min_{\pi \in \Pi_{\rho^{\mathrm{TSP}}}} \mathbb{E}\left[\sum_{k'=k}^{K} C(s_{k'}, X_{k'}^{\pi}(s_k')) \right] = v_{k+1}^{\mathbb{F}}(s_k, a).$$

The reversal of expectation and minimization that produces the middle inequality is a result of the use of perfect information in the construction of $v_{k+1}^{\mathbb{I}}(s_k, a)$. 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

$$\hat{z}(s_j, a_{j,j'}) \leq \hat{z}(s_c, a_{c,j'}), \tag{40}$$

then we are done.

Writing the penalties explicitly and somewhat abusing notation for epoch indices, inequality (40) is equivalent to

$$v_{j+1}^{\mathbb{F}}(s_j, a_{j,j'}) - v_{j+1}^{\mathbb{I}}(s_j, a_{j,j'}) \leq v_{c+1}^{\mathbb{F}}(s_c, a_{c,j'}) - v_{c+1}^{\mathbb{I}}(s_c, a_{c,j'}). \tag{41}$$

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 (41) holds, meaning the triangle inequality also does in the presence of penalties. $\qquad\square$

By Theorem 3, because the optimal policy for the penalized perfect information problem is AC, we can write its objective function as

$$\mathbb{E}\left[\min_{\pi \in \Pi} \sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) + \hat{z}(s_k, X_k^\pi(s_k))\right] = \mathbb{E}\left[\min_{\pi \in \Pi^{\mathrm{AC}}} \sum_{k=0}^{K} C(s_k, X_k^\pi(s_k)) + \hat{z}(s_k, X_k^\pi(s_k))\right]$$

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

Restricting our search to the set of AC policies is especially convenient, because there are significantly fewer charging decisions to search over 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. 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 $s_{k(j')}$:

$$\min_{\pi \in \Pi_{\rho^{\mathrm{TSP}}}} \mathbb{E}\left[\sum_{k=k(j')}^{K} C(s_k, X_k^\pi(s_k))\right] \leq \min_{\pi \in \Pi_{\rho^{\mathrm{TSP}}}} \mathbb{E}\left[\sum_{k=k(cj')}^{K} C(s_k, X_k^\pi(s_k))\right]. \tag{42}$$

The left-hand side of (42) 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 (42) 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 (23)-(26). By the definition of this action space, the only actions exclusively available downstream from state $s_{k(cj')}$ are those in equation (24) 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

Table 2: Disaggregated results showing average objective values (in min) by instance size (number of customers).

	4 (Penalty)	10 w/ 2 public CSs	10 w/ 4 public CSs	20	30	40
PI+Penalty	255.46	-	-	-	-	-
PI	237.12	516.53	452.04	-	-	-
Opt Static	343.69	545.13	469.51	-	-	-
Private Only	-	700.02	608.29	-	-	-
PreOpt	-	545.43	469.47	-	-	-
TSP Static	-	565.53	480.84	777.81	1020.23	889.01
OSM	-	630.84	554.41	832.19	1149.51	980.05
PostTSP	273.47	554.87	471.21	729.45	954.55	854.98

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 (42) holds. $\qquad\square$

F Disaggregated Results of Computational Experiments

Table 2 contains disaggregated results for the computational experiments described in §7. The results are divided by instance size. Values are averages over all realizations of uncertainty for all instances for a given size.