Electric Vehicle Routing with Public Charging Stations ***appendices***

Appendix 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:

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

$$a^i = i_k, a^q \in \Bigg\{ \tilde{q} \in \mathcal{Q} \,\Bigg|\, \tilde{q} > q_k \wedge \big((\exists c \in \mathcal{C}, \exists j \in \bar{\mathcal{N}}_k : \tilde{q} \geq e_{i_k j} + e_{jc}) \vee (\bar{\mathcal{N}}_k = \emptyset \wedge \tilde{q} \geq e_{i_k 0}) \big) \Bigg\}, \quad i_k \in \mathcal{C} \tag{35}$$

$$a^i \in \bar{\mathcal{N}}_k, a^q = q_k - e_{i_k a^i}, \quad (\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) \tag{36}$$

$$a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i}, \quad q_k \geq e_{i_k a^i} \Bigg\}. \tag{37}$$

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

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

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

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

Appendix B: Proof of Proposition 1

We begin by repeating the statement for Proposition 1:

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

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

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. The construction strategy for π^{AC} in the proof requires knowledge of these immediate departures a priori, so it is therefore only valid in the context of static policies. We note that this proof holds under any information filtration.

Appendix C: Proof of Proposition 2

We begin by repeating the statement for Proposition 2:

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

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

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

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

Appendix D: Defining Action Space for FRVCPs

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

$$\begin{aligned}
\mathcal{A}^{\mathrm{AC}}(s_k, \rho) = \Big\{ & (a^i, a^q) \in \{n^\star \cup \mathcal{C}\} \times [0, Q] : \\
& a^i = i_k, a^q = q_k, \\
& \quad i_k \in \mathcal{C}' \wedge \psi_{i_k} < z_k && (38) \\
& a^i = i_k, a^q \in \{\tilde{q} \in \mathcal{Q} | \tilde{q} > q_k \wedge (\exists c \in \mathcal{C} : \tilde{q} \geq e_{i_k n^\star} + e_{n^\star c})\}, \\
& \quad i_k \in \mathcal{C} \wedge z_k \leq \psi_{i_k} && (39) \\
& a^i = n^\star, a^q = q_k - e_{i_k a^i}, \\
& \quad (\exists c \in \mathcal{C} : a^q \geq e_{a^i c}) \wedge (i_k \in C \Rightarrow q_k > q_{k-1}) && (40) \\
& a^i \in \mathcal{C} \setminus \{i_k\}, a^q = q_k - e_{i_k a^i}, \\
& \quad q_k \geq e_{i_k a^i} \wedge (i_k \in C \Rightarrow q_k > q_{k-1}) \Big\}. && (41)
\end{aligned}$$

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

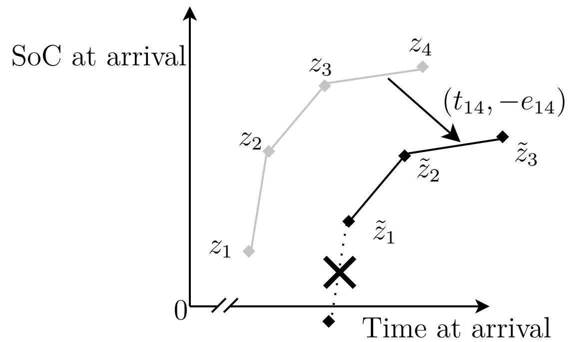

Figure 13 **An example of shifting the SoC function as we extend the label along the edge from customer node 1 to CS node 4d in Figure 5. The SoC function for node 1 (in gray) is translated by $(t_{14}, -e_{14})$. The resulting SoC function for the label at node 4d (in black) contains one fewer supporting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

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

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

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

To accommodate time-dependent waiting times, we make additional adjustments to the SoC function when extending a label to a CS node, such as to node 4d. We want the supporting points in the SoC function to reflect the time at which the vehicle enters service at the CS. To do so, after the initial translation (depicted in Figure 13), we shift the SoC function supporting points again according to the underlying wait time (either known or expected). Define the function $w : (\mathbb{R}_{\geq 0} \times \mathcal{C}) \rightarrow \mathbb{R}_{\geq 0}$ that specifies the (known or expected) waiting time that the EV incurs if it arrives to some CS c at some time t. In the case of known waiting times, functions $w(t, c')$ may not be continuous for a given CS c' (see Figure 4, right), so we cannot represent the resulting SoC function as continuous. We group the supporting points based on discontinuities in $w(t, c')$ and create a new label for each group.

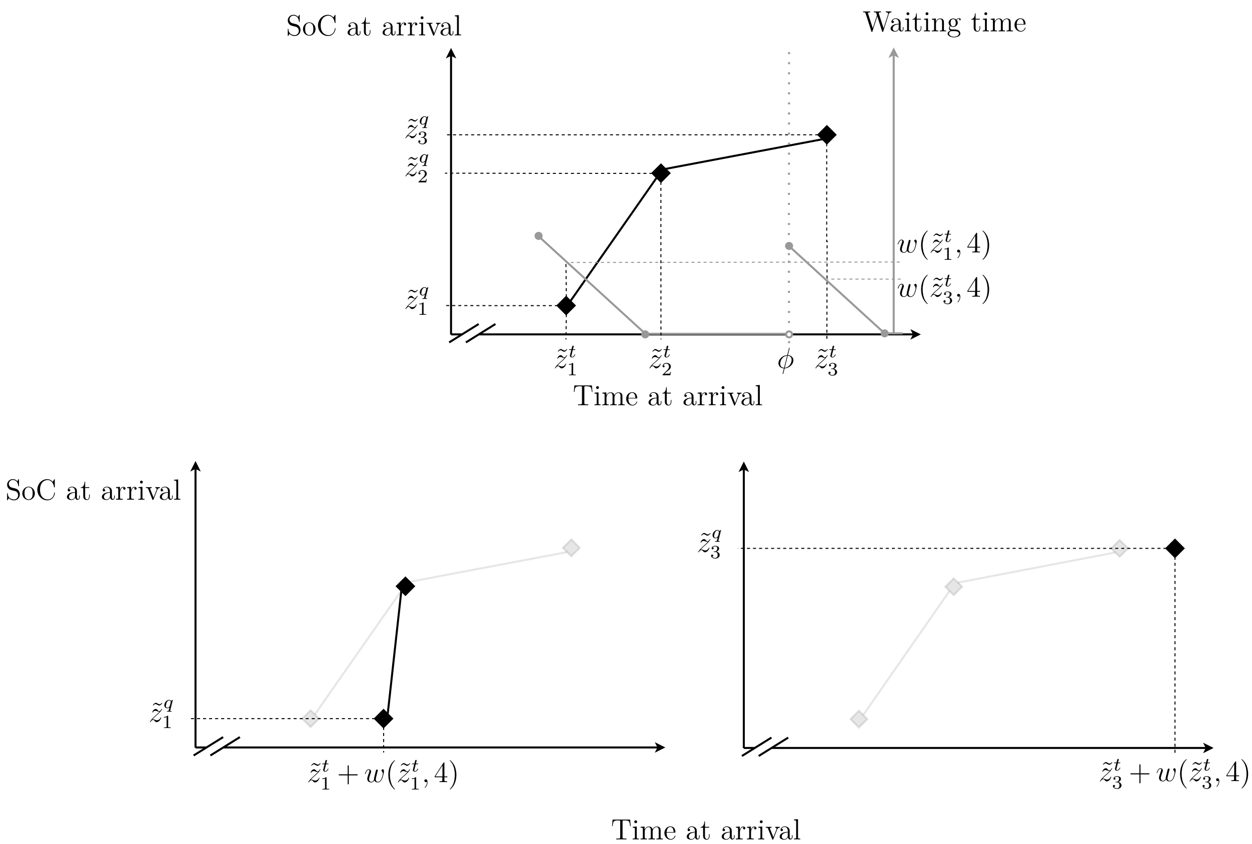

Figure 14 **Depiction of handling time-dependent waiting times. In the top graph, we have the resulting SoC function after the initial translation from node 1 to node 4d depicted in Figure 13. This is superimposed over the wait-time function $w(t, 4)$, plotted in gray. The supporting points for the SoC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

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

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

Appendix F: Information Penalties

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

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

The form of the information penalty we use is $z(s_k, a) = \mathbb{E}\left[V_{k+1}(s_k, a)|\mathscr{F}_k\right] - \mathbb{E}\left[V_{k+1}(s_k, a)|\mathscr{I}_k\right]$, where $V_{k+1}(s_k, a)$ is the value of being in the pre-decision state s_{k+1} reached by choosing action a from state s_k. The penalty captures the difference in the expected cost-to-go under the natural and perfect information filtrations. The form of this penalty aligns with that of Theorem 2.3 (and Proposition 2.2) of Brown, Smith, and Sun (2010), which promises strong duality. Strong duality guarantees that the optimal objective value of the penalized perfect information problem (42) will be equal to the objective value of the optimal non-anticipative policy. In practice, however, the values $\mathbb{E}\left[V_{k+1}(s_k, a)|\mathscr{F}_k\right]$ and $\mathbb{E}\left[V_{k+1}(s_k, a)|\mathscr{I}_k\right]$ are unknown. To approximate them, we follow an approach suggested in Brown, Smith, and Sun (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 $\mathbb{G}$: $v_{k+1}^{\mathbb{G}}(s_k, a) = \mathbb{E}\left[\sum_{i=k+1}^{K} C\left(s_i, X_i^{\pi(s_{k+1},\mathbb{G})}(s_i)\right)\middle| s_k, a\right]$. For our estimating policy, we use the TSP static policy (see §5.1.2). Then we may write our penalty explicitly as

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

The objective for the penalized PI problem with our approximation is

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

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

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

Proof. See §F.2.

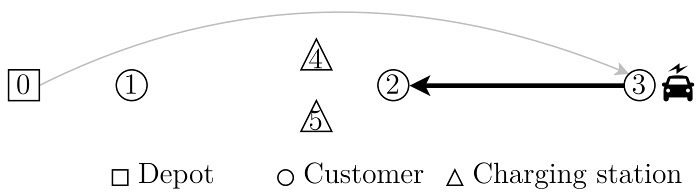

Figure 15 A vehicle at customer 3 in the beginning of epoch one. The vehicle must visit customers 2 and 1 before returning to the depot, but before it can visit customer 1, it must first charge. We illustrate the construction of information penalties using the action associated with the bolded arrow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

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

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

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

For an example of the construction of information penalties, let us consider Figure 15 with the vehicle in state $s_1 = (t_{0,3}, 3, Q - e_{0,3}, \{2,1\}, 1)$ (note the omission of queue memories). We assume CSs 4 and 5 are identical, meaning they have the same charging technology and number of chargers. Further, we assume that $t_{2,4} = t_{2,5}$ and $t_{4,1} = t_{5,1}$ (likewise for the energy to traverse these arcs). We compute a penalty for each action in the action space $\mathcal{A}(s_1)$, which consists of relocation actions to customer 2 and charging stations 4 and 5 (relocating to nodes 0 and 1 is energy infeasible). Abusing notation slightly, we have $\mathcal{A}(s_1) = \{a_2 \equiv (2, q_1 - e_{3,2}); a_4 \equiv (4, q_1 - 3_{3,4}); a_5 \equiv (5, q_1 - e_{3,5})\}$. In this example, we will illustrate the computation of the penalty $\hat{z}(s_1, a_2)$ corresponding to the action a_2 in which the EV relocates to customer 2.

First, from the post-decision state $s_1^{a_2}$, we sample realizations of queue dynamics at CSs 4 and 5. For simplicity, let us assume we are conducting a single sample denoted by $\omega \in \Omega$. We realize the (deterministic) exogenous information $W_2 = (t_1 + t_{3,2}, 1) \in \mathcal{I}(s_1^{a_2})$ and transition to state $s_2 = (t_{0,3} + t_{3,2}, 2, Q - e_{0,3} - e_{3,2}, \{1\}, 1)$. From this state, we wish to construct TSP Static policies $\pi(s_2, \mathbb{F})$ and $\pi(s_2, \mathbb{I})$ for use in $v_2^{\mathbb{F}}(s_1, a_2)$ and $v_2^{\mathbb{I}}(s_1, a_2)$, respectively. Per §5.1.2, the CL sequence followed by the vehicle will be the same under both filtrations, so we determine it first. To do so, we solve a single iteration of the outer minimization of equation (15). This finds the shortest Hamiltonian path from customer

2, through the remaining customers, terminating at the depot, which is the sequence $\rho = (2,1,0)$. Then, given ρ, we solve a single iteration of the inner minimization to establish the fixed route for the TSP static policies: we solve the FRVCP-N on ρ to construct the fixed route we call $p^{\mathbb{F}}$ and its corresponding policy $\pi(s_2,\mathbb{F}) = \pi(p^{\mathbb{F}})$, and we solve the FRVCP-P on ρ to construct the fixed route we call $p^{\mathbb{I}}$ with corresponding policy $\pi(s_2,\mathbb{I}) = \pi(p^{\mathbb{I}})$. For the former, let us assume that the expected waiting time at CS 4 is 40 min, and the expected waiting time at CS 5 is 45 min. This leads to the fixed-route solution $p^{\mathbb{F}} = ((2,q_2),(4,q_2 - e_{2,4}),(4,\tilde{q}),(1,\tilde{q} - e_{4,1}),(0,\tilde{q} - e_{4,1} - e_{1,0}))$, which includes a stop to charge at CS 4 to charge level $\tilde{q} = \min\{q \in \mathcal{Q}'\}$ where $\mathcal{Q}' = \{q \in \mathcal{Q}:\ q \geq e_{4,1} + e_{1,0}\}$. The cost of $p^{\mathbb{F}}$ we denote $\tau(\pi(p^{\mathbb{F}})) = t_{2,4} + 40 + \bar{u}(q_2 - e_{2,4},\tilde{q}) + t_{4,1} + t_{1,0}$. For the FRVCP-P we proceed similarly, except now we have access to ω, which grants us knowledge of the queue dynamics at CS 4 and 5 at all points in time. Say we know the wait time at CS 4 will actually be 20 min, and the wait time at CS 5 will be 5 min. Then the solution to the FRVCP-P is the fixed route $p^{\mathbb{I}} = ((2,q_2),(5,q_2 - e_{2,5}),(5,\tilde{q}),(1,\tilde{q} - e_{5,1}),(0,\tilde{q} - e_{5,1} - e_{1,0}))$ with corresponding cost $\tau(\pi(p^{\mathbb{I}})) = t_{2,5} + 5 + \bar{u}(q_2 - e_{2,5},\tilde{q}) + t_{5,1} + t_{1,0}$. The values $v_2^{\mathbb{F}}(s_1,a_2)$ and $v_2^{\mathbb{I}}(s_1,a_2)$ are then equal to the average of the route costs associated with $\pi(p^{\mathbb{F}})$ and $\pi(p^{\mathbb{I}})$, respectively, over samples from Ω (of which there is only one in this example). Thus, we have $\hat{z}(s_1,a_2) = v_2^{\mathbb{F}}(s_1,a_2) - v_2^{\mathbb{I}}(s_1,a_2) = \mathbb{E}\left[\tau(\pi(p^{\mathbb{F}}))\right] - \mathbb{E}\left[\tau(\pi(p^{\mathbb{I}}))\right] = 40 - 5 = 35$, so the penalized cost of choosing action a_2 from state s_1 is $C(s_1,a_2) + \hat{z}(s_1,a_2) = t_{3,2} + 35$. The value of the penalty represents the benefit of using advanced information in decision making, capturing the difference in expected costs-to-go $\mathbb{E}\left[V_2(s_1,a_2)|\mathscr{F}_1\right]$ and $\mathbb{E}\left[V_2(s_1,a_2)|\mathscr{I}_1\right]$.

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

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

F.1. Experiments with Information Penalties

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

Because the reaching algorithm used to solve the penalized FRVCP (the inner minimization of (45)) enumerates all reachable states along a fixed CL sequence, we must be mindful of instance size in these experiments. To ensure tractability, we construct an instance with four customers and two extradepot CSs.

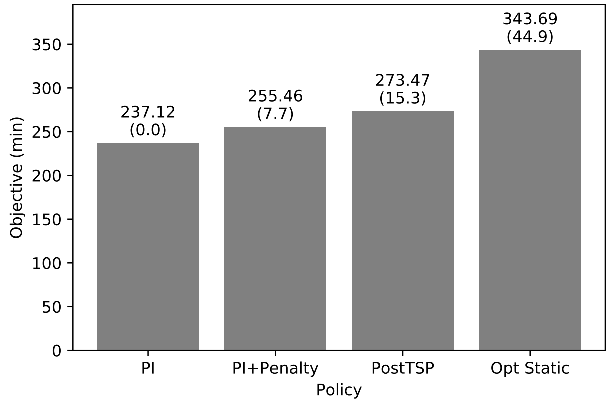

Figure 16 Comparing our best dynamic and static non-anticipative policies to the dual bounds afforded by the value of the optimal policy with perfect information and the value of the optimal policy with penalized access to perfect information.

Note. Bar labels are average objective achievement over 250 samples of uncertainty with percent difference from the PI bound in parentheses.

Further, we limit the set of chargeable battery states $\mathcal{Q}$ to the charging function breakpoints and multiples of 25% ($\mathcal{Q} = \{0, 0.25Q, 0.5Q, 0.75Q, 0.85Q, 0.95Q, Q\}$). Despite these restrictions, the computational effort required to solve just one realization of uncertainty with information penalties is almost ten minutes. This is in contrast to the negligible computation time (milliseconds) required to establish the perfect information bound for this instance, as well as execute all other routing policies.

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

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

F.2. Proof of Proposition 4

We begin by repeating the statement of Proposition 4:

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

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

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

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

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

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

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

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

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

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

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

then we are done.

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

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

Notice, however, that each term represents an expected cost-to-go from node j'. The terms on the left-hand side represent costs-to-go from node j' after traveling directly from j, while terms on the right-hand side represent costs-to-go after first balking CS c. Notice also that, for a given filtration, the cost-to-go from node

j' cannot be better after balking at CS c than if having traveled directly. To prove this is the case, we refer the reader to Lemma 1.

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

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

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

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

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

Proof. 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 $s_{k(j')}$:

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

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

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

Appendix G: Proof of Proposition 3

We begin by repeating the statement for Proposition 3:

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

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

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

Appendix H: Disaggregated Results of Computational Experiments

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

Appendix I: Tabular Summary of Notation

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

Table 5 **Disaggregated objective values over testbed of instances, including CV bound and the objective value under the private-only recharging strategy.**

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

Values in minutes. "inf" entries (Private-only) were infeasible. We set a solver time limit of 3 hrs. Throughout, asterisks (*) indicate optimality. For PI, this implies optimal solutions to all 50 realizations of uncertainty for the instance. For non-optimal solutions, Optimal Static, PreOpt, and QPreOpt use the solver's best feasible solution; PI uses the solver's best bound.

Table 6 Notation used in the manuscript, presented in order of appearance

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

*These symbols may also appear with a superscript a, referring to their value in the post-decision state after taking action a in epoch k

References

Brown DB, Smith JE, Sun P, 2010 *Information relaxations and duality in stochastic dynamic programs. Operations research* 58(4-part-1):785–801.

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

Froger A, Mendoza JE, Jabali O, Laporte G, 2019 *Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Computers & Operations Research* 104:256–294.