Adaptive Routing and Recharging Policies for Electric Vehicles

Timothy M. Sweda, Irina S. Dolinskaya, Diego Klabjan

Department of Industrial Engineering and Management Sciences

Northwestern University

2145 Sheridan Rd.

Evanston, Illinois 60208, USA

E-mail: tsweda@u.northwestern.edu, dolira@northwestern.edu, d-klabjan@northwestern.edu

Abstract

Planning a trip with an electric vehicle requires consideration of both battery dynamics and the availability of charging infrastructure. Recharging costs for an electric vehicle, which increase as the battery's charge level increases, are fundamentally different than refueling costs for conventional vehicles, which do not depend on the amount of fuel already in the tank. Furthermore, the viability of any route requiring recharging is sensitive to the availability of charging stations along the way. In this paper, we study the problem of finding an optimal adaptive routing and recharging policy for an electric vehicle in a grid network. Each node in the network represents a charging station and has an associated probability of being available at any point in time or occupied by another vehicle. We present an efficient algorithm for finding an optimal a priori routing and recharging policy as well as two heuristic methods for finding adaptive policies – one with adaptive recharging decisions only and another with both adaptive routing and recharging decisions. We conduct numerical experiments to demonstrate the empirical performance of our solutions.

Keywords: electric vehicles; adaptive routing; recharging policies; dynamic programming

1 Introduction

Battery electric vehicles (EVs) have become a practical and affordable option in recent years for environmentally conscious drivers looking to reduce their carbon footprint. EVs are powered solely by electricity and recharge by plugging into an outlet or charging station. This feature offers the potential for significant fuel cost savings as well as a host of other benefits, such as fewer greenhouse gas emissions, reduced dependence on foreign oil, and improved power systems management (if vehicle-to-grid services are enabled). Despite the numerous advantages that EVs have over conventional gasoline-powered vehicles, range anxiety – the worry that an EV's range is insufficient for a driver's commuting needs – remains a chief concern among many potential purchasers [7]. Without the ability to easily recharge away from home, EV owners are restricted primarily to short-distance trips. The expansion of public charging infrastructure to increase opportunities for EV drivers to recharge their vehicles outside their homes has been one popular method for alleviating range anxiety and encouraging greater EV adoption. However, since each charging station can usually only recharge one or two vehicles at a time, and charge times can be on the order of hours, a driver who arrives at a fully occupied station may incur significant inconvenience (e.g., a long wait time) if no other nearby charging station is available. Thus, EV drivers can greatly benefit from taking into account charging station availability and anticipating wait times at the stations while planning their routes, which is the focus of this paper.

We study an optimal EV routing problem in a grid network where trips may require multiple recharging stops along the way. Since the availability of charging stations is a critical factor in the planning of long-distance routes, an EV driver must not only select which path to take to arrive at the destination as quickly as possible, but also decide where to recharge and what to do in case a desired charging station is unavailable (i.e., wait or seek an alternative station). The uncertainty of charging station availability and wait times within the network as well as the driver's ability to adaptively make routing and recharging decisions are unique and critical features of our problem. Furthermore, since the availability of each station may differ, the selection of stopping locations must be part of the routing decision. We consider the problem on a given network with charging stations positioned at every node, each with a corresponding probability of being available and expected waiting time until becoming available (known a priori to the driver). Our goal is to determine an adaptive routing and recharging policy that minimizes the sum of all traveling, waiting, and recharging costs. We assume that whenever the vehicle stops to recharge, it incurs a fixed stopping cost, a charging cost based on the total amount it recharges, and an additional cost when the battery becomes overcharged.

In this paper, we develop and analyze a variety of models depending on the amount and timing of information available to the EV driver while traveling. In our first model, an a priori model, we find an optimal route that minimizes the total expected cost. Our second set of models considers a setting in which information about a station's availability is revealed to the driver upon arriving at that location, and the driver decides at that time whether or not to stop. This additional level of driver adaptivity in response to the real-time data results in two solution methods for adaptive decision making: (1) the driver is committed to a path corresponding to the optimal a priori policy, but can change the charging locations along the way; and (2) the driver can adapt both the recharging locations and path in response to the realized availability of charging stations. A numerical study demonstrates the performance and solution quality of our algorithms using simulated urban charging infrastructure networks.

This paper is the first in the literature to consider adaptive routing and recharging (or refueling) for range-constrained vehicles. It is also the first to implement two features together that are quite unique to EVs: overcharging costs and uncertain charging station availability. Thus, the main contributions of this paper are: (i) properties of optimal adaptive and a priori recharging policies that consider EV overcharging characteristics and uncertain charging station availability; (ii) efficient solution procedures for obtaining a priori and adaptive routing and recharging policies in a grid network; (iii) models capturing and analyzing various levels of adaptive decision making and information timing; and (iv) an illustrative example and numerical study that establish the value of adaptive decision making in a variety of urban settings.

The remainder of the paper is organized as follows. Section 2 provides an overview of the existing literature related to EV recharging as it pertains to routing decisions. Section 3 describes the general problem setting and provides a motivating example as well as an overview of our solution approach. In Section 4 we present efficient solution methods to find an optimal path and recharging locations for an a priori setting, first looking at a fixed path and then in a grid network. In Section 5, we study two versions of an adaptive problem where (1) the vehicle commits to a path a priori and can only adapt its charging locations along the path; and where (2) the driver updates the path and recharging nodes en route in response to realized station availability. Subsequently, a numerical study is exhibited in Section 6 to demonstrate the performance of the solution algorithms presented in the earlier sections. Finally, Section 7 presents the conclusion of this paper.

2 Literature review

The vehicle refueling problem, where a driver must decide at which nodes to refuel as well as how much to refuel in order to minimize the total cost of fuel, has been well studied. It is shown in [6] and [10] that the optimal refueling policy along a fixed path can be solved easily with dynamic programming when fuel prices at each node are static and deterministic. For such a problem, the optimal decision at each node is always one of the following: do not refuel, refuel completely, or refuel just enough to reach the next node where refueling occurs. An algorithm for simultaneously finding the optimal path and refueling policy in a network is detailed in [11], and some combinatorial properties of the optimal policies are explored in [12]. Specifically, it is proven that the problem of finding all-pairs optimal refueling policies reduces to an all-pairs shortest path problem that can be solved in polynomial time. However, all of the aforementioned analyses only consider fuel costs and not stopping or other costs. We include these additional costs in our analysis since they can comprise a significant portion of the total travel cost for an EV and therefore can influence optimal policies.

Several existing models have expanded on the vehicle refueling problem by introducing costs for stopping to refuel and traveling to refueling stations. A generic model for vehicle refueling is presented in [17] that attempts to capture such aspects, penalizing longer routes and routes with more refueling stops. Similar to other papers that study the vehicle refueling problem, it assumes that fuel prices at each station are static and deterministic. An analogous model is proposed in [19], which seeks to minimize both the fuel cost and travel distance simultaneously. Approaches for finding optimal refueling policies when fuel prices are stochastic are presented in [8] and [18]. In [8], a forecasting model for predicting future fuel prices is used to generate parameters for a deterministic mixed integer program, and in [18], a dynamic programming framework is presented that is designed to grant drivers greater autonomy to select the stations where they refuel. These models are difficult to solve analytically, and the authors develop heuristics for obtaining reasonable solutions. In addition, just like the other models of the vehicle refueling problem, these ones do not include any costs that are analogous to battery overcharging costs for EVs, which can be an important component [13, 16]. The authors in [20] address this issue by introducing generalized charging cost functions, however some restrictive assumptions are required in order to perform insightful analysis. A more specific yet tractable overcharging cost function is analyzed in [21], and optimal recharging policies along a fixed path are determined.

Overcharging costs, incurred when an EV's battery is charged near its maximum capacity, are important to consider when creating EV recharging policies for a number of reasons. First, recharging an EV battery

while it is already at a high state of charge takes place at a slower rate than when it is more depleted, and storing high levels of charge for prolonged periods of time can shorten the lifespan of the battery. A couple of models describing this relation can be found in [13] and [16]. In addition, overcharging causes battery degradation due to greater stresses from being charged near full capacity and excess heat generated during recharging. In the refueling problem for conventional vehicles, the only main disadvantage of traveling with a full tank of fuel is the limited ability to take advantage of lower fuel prices further along the route. Thus, optimal solutions tend to favor filling large quantities and making fewer stops (assuming that stopping costs are considered), but the opposite is often true for optimal EV recharging policies.

To solve the problem of finding a path for an EV within a network with recharging considerations, one thread of research has taken an entirely different approach: having vehicles recharge via regenerative braking rather than by recharging at stations along their paths. As an EV decelerates, it can recapture some of its lost kinetic energy as electrical energy, which can then be used to recharge the battery. It is therefore possible in some cases for an EV's state of charge to *increase* while traveling rather than decrease, such as when the vehicle is coasting and braking downhill. In [1], the authors model the problem of finding the most energy-efficient path for an EV in a network as a shortest path problem with constraints on the charge level of the vehicle, such that the charge level can never be negative and cannot exceed the maximum charge level of the battery. Edge weights are permitted to be negative to represent energy recapturing from regenerative braking, yet no negative cycles exist. A simple algorithm for solving the problem is provided, and more efficient algorithms are presented in [2] and [15]. It is shown in [2] that the battery capacity constraints can be modeled as cost functions on the edges, and a transformation of the edge cost functions permits the application of Dijkstra's algorithm. The approach described in [15] avoids the use of preprocessing techniques so that edge costs can be calculated dynamically, and it achieves an order of magnitude reduction in the time complexity of the algorithm from [1]. In practice, however, the amount of energy recovered by regenerative braking is insignificant when compared with the amount that must be recharged at charging stations, and these papers do not model recharging decisions at nodes. Consequently, they also do not capture overcharging costs considered in the presented work.

All of the aforementioned models assume that there is no downtime for refueling or recharging stations, and that a vehicle never needs to wait to use a station. Although this may be valid for gasoline stations, which can typically accommodate multiple vehicles at the same time and often have negligible wait times, it is not necessarily the case for electric charging stations. Most charging stations have only one or two plugs, and due to the long time that it takes to recharge a vehicle, the availability of a charging station can become

an issue when multiple vehicles are present. To the best of the authors' knowledge, there is no other work in the literature that addresses this problem of finding an optimal recharging policy when recharging station availability is stochastic. However, there are several papers that consider routing problems with stochastic elements, and we refer the reader to [4] for a review of the existing literature. The body of work that is perhaps most relevant to the study presented in this paper involves anticipatory routing, where the goal is to anticipate information before it becomes available and minimize the total expected travel cost. In [22], a vehicle travels from an origin to a destination, and along the way customers may request service according to known distributions. Customer requests may be realized at any point along the vehicle's path, and the vehicle must respond to all requests. An optimal route therefore is not the shortest path between the origin and destination, but instead includes detours to approach potential customers in the event that they request service. A similar model in which congestion along arcs is the anticipated information is studied in [23], and in both papers, structural properties of optimal anticipatory policies are derived. However, the models only consider routing decisions and not refueling decisions.

3 Problem setting, motivation and solution approach

In this section we detail the setting of our adaptive routing and recharging problem for electrical vehicles. We provide a motivating example to demonstrate the potential benefits of adaptive routing and recharging and then outline the solution approaches to be expanded on in subsequent sections.

3.1 Problem setting

Consider an EV with zero initial charge level that must travel within a network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ from origin node o to destination node n, where o and n are in the node set $\mathcal{N}$, and $\mathcal{E}$ is the set of edges. Every node $i \in \mathcal{N}$ has a charging station, however the availability of each station is uncertain (e.g., due to being occupied by another vehicle) and is denoted by a random variable A_i. When the vehicle arrives at node i and finds the station available (i.e., $A_i = 1$), it may begin recharging immediately. In the event that the charging station unavailable (i.e., $A_i = 0$), the driver must wait W_i time interval until the station becomes available before recharging, where W_i is also a random variable. We assume that for each $i \in \mathcal{N}$, the driver knows the probability of station i being available, $P(A_i = 1)$, and the expected waiting time when it is unavailable, $E[W_i|A_i = 0]$, a priori. Then, the values of these random variables for a given node are realized when the EV arrives at the node and observes the state of the charging station.

As the driver selects a path to travel from o to n, he or she must decide which nodes to visit, whether or not to stop and recharge at each visited node (station), and how much to recharge at each stop. The vehicle's charge level can never exceed the maximum capacity of the battery, q_{max}, and it can never drop below zero. We also do not allow the vehicle to discharge energy back to the grid. We let q_i denote the charge level of the vehicle when it arrives at node i, and for any pair of nodes i and j such that $(i, j) \in \mathcal{E}$, we let $t_{ij} > 0$ and $h_{ij} > 0$ denote the time and amount of charge, respectively, required to traverse (i, j). Then the set of feasible charging amounts at node i prior to traveling along an edge (i, j), which we denote by $R_{ij}(q_i)$, is

$$R_{ij}(q_i) = \left[(h_{ij} - q_i)^+, \ q_{max} - q_i \right]. \tag{1}$$

We assume that $h_{ij} \leq q_{max}$ for all $(i, j) \in \mathcal{E}$ to ensure feasibility.

Each time the vehicle stops to recharge, it incurs a fixed *stopping cost s*. The vehicle also incurs a *recharging cost* at a rate of $\gamma \geq 0$ per unit of energy recharged plus an additional *overcharging cost* when its charge level exceeds αq_{max}, where $0 < \alpha < 1$. This threshold αq_{max} represents a point at which the charging voltage reaches its maximum value and the charging current begins to decrease. Thus, the overcharging cost takes into account the additional time per unit of energy recharged (caused by the decreasing current). We denote the overcharging cost as $f(z)$, where $z \in [0, (1 - \alpha)q_{max}]$ is the amount by which the vehicle's charge level exceeds αq_{max} after recharging, and $f(\cdot)$ is a convex and increasing function with $f(0) = 0$. Such properties are reflective of actual battery dynamics that make it undesirable to regularly overcharge the battery. If the vehicle's charge level (q) already exceeds αq_{max} when it stops to recharge, we discount the overcharging cost by $f(q - \alpha q_{max})$. Therefore, if we let $c(r, q)$ denote the cost of recharging amount of energy r when the vehicle's initial charge level is q, then we have

$$c(r, q) = \gamma r + \left[f \left((q + r - \alpha q_{max})^+ \right) - f \left((q - \alpha q_{max})^+ \right) \right]. \tag{2}$$

See Figure 1 for an example of $c(r, 0)$. We assume that all charging stations have identical cost function $c(., .)$, which is a reasonable assumption since most public charging stations in existence today have similar hardware configurations (most recharge at 220 volts, with the exception of a few "fast charging" stations that recharge at 440 volts [3]). Furthermore, regional variations in electricity rates are minimal.

We formulate our problem using a dynamic programming approach as follows.

Objective: Find a minimum-cost routing and recharging policy from an origin node o to a destination node n. The total cost is measured in units of time and is equal to the sum of all travel time, waiting time for
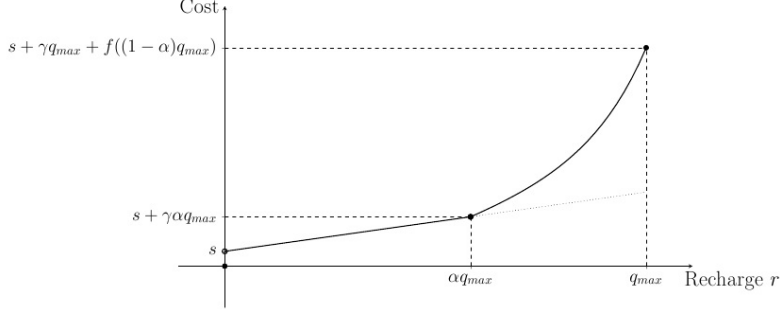
Figure 1: Sample illustration of $c(r, q)$ for $q = 0$ **??**

available charging stations, and charging costs (i.e., stopping, recharging and overcharging costs).

State space: The state space, which we denote by $\mathcal{S}$, is defined as

$$\mathcal{S} = \{(i, q_i, a_i) : i \in \mathcal{N}, \; q_i \in [0, q_{max}], \; a_i \in \{0, 1\}\}.$$

Each state consists of three components: the current vehicle position at node i, the vehicle's charge level q_i upon arrival at the node, and the realized availability of the charging station at the node, denoted by a_i.

Action space: We denote the action space by

$$\mathcal{A}_{(i, q_i)} = \{(j, r_i) : (i, j) \in \mathcal{E}, \; r_i \in R_{ij}(q_i)\},$$

where each action consists of the next node to visit, j, along with the amount to recharge at the current node, r_i. The next node must be adjacent to the current node, and the recharging amount must be feasible (where $R_{ij}(q_i)$ is as defined in (1)) to allow the vehicle to reach the next node.

Value function: We let $V(i, q_i, a_i)$ denote the value function, which represents the minimum expected cost of traveling to the destination n from node i given that the current state is (i, q_i, a_i). We also let $I_{r_i > 0}$ denote the indicator function that equals 1 if $r_i > 0$ (i.e., if the vehicle stops to recharge at node i) and 0 otherwise. Then the value function can be defined recursively as

$$V(i, q_i, a_i) = \min_{(j, r_i) \in \mathcal{A}_{(i, q_i)}} \{t_{ij} + (s + E[W_i | A_i = a_i] + c(r_i, q_i)) I_{r_i > 0} + E_{A_j}[V(j, q_i + r_i - h_{ij}, A_j)]\}$$

$$= \min_{(j, r_i) \in \mathcal{A}_{(i, q_i)}} \{t_{ij} + (s + (1 - a_i) E[W_i | A_i = 0]) I_{r_i > 0} + c(r_i, q_i) +$$

$$[P(A_j = 1)] V(j, q_i + r_i - h_{ij}, 1) + [1 - P(A_j = 1)] V(j, q_i + r_i - h_{ij}, 0)\} \quad (3)$$

8

since $E[W_i | A_i = 1] = 0$ and $c(0, q_i) = 0$ for any value of q_i. We set $V(n, \cdot, \cdot) = 0$ and seek to evaluate $E[V(o, 0, A_m)]$, the minimum expected total cost of traveling from o to n when the vehicle's initial charge level is zero.

3.2   Motivation for adaptive decision making

To see the benefits of adaptive routing and recharging decisions, consider the example shown in Figure 2. The road network consists of nine nodes arranged in a square grid, with the origin $(0, 0)$ and destination $(2, 2)$ at opposite corners, and $q_{max} = 2h$, where h is the amount of charge necessary to traverse any edge. The label next to each node on the figure corresponds to the expected wait time when the EV finds the station unavailable upon arrival (i.e., $E[W_i | A_i = 0]$), and the availability probabilities for each node are 0.5 (i.e., $P(A_i = 1) = 0.5$). Furthermore, the nodes without any labels (those shaded in light grey) do not have any recharging facilities, that is, $P(A_i = 1) = 0$ and $E[W_i | A_i = 0] = \infty$. For illustration purposes, we consider only waiting costs and neglect any stopping, recharging, overcharging, and traveling costs.

In this setting, an optimal a priori policy, shown in Figure 3(a), is to recharge $2h$ at the origin and $2h$ at node $(0, 2)$ with a total expected waiting cost of $0.5 \cdot (0) + 0.5 \cdot (2) = 1$. Now consider an adaptive policy where we allow the driver to dynamically change the recharging decisions, but we fix the path to an a priori optimal path, as shown in Figure 3(b). Thus, we allow the vehicle to recharge at any node along the path in response to the observed station availability upon arrival at the node. The optimal policy in such a case is to recharge $2h$ at the origin, and if the charging station at node $(0, 1)$ is unavailable when the vehicle arrives, then the vehicle should recharge 0 at $(0, 1)$, $2h$ at $(0, 2)$, and 0 at $(1, 2)$. Otherwise, if the driver finds station at node $(0, 1)$ to be available, then the vehicle should recharge h at node $(0, 1)$. Then, the EV does not have to recharge at node $(0, 2)$ and will not do so unless that station is also available upon arrival. That is, the driver will recharge h at node $(0, 2)$ if the station there is available (and then 0 at $(1, 2)$) or will charge h at $(1, 2)$ and 0 at $(0, 2)$ if the station at node $(0, 2)$ is unavailable. The cost of such a policy is the same as that of the optimal a priori policy when the station at node $(0, 1)$ is unavailable, yet when that station is available, then the only scenario in which the expected waiting cost is nonzero is if both the stations at $(0, 2)$ and $(1, 2)$ are unavailable, in which case the expected waiting cost is 2. The overall expected waiting cost of the optimal policy with adaptive recharging decisions is therefore $0.5 \cdot (1) + 0.5 \cdot (0.75 \cdot (0) + 0.25 \cdot (2)) = 0.75$, which is 0.25 less than the cost of the optimal a priori policy.

In the model where both adaptive routing and recharging decisions are allowed, note that the optimal policy when the station at node $(0, 1)$ is available is to recharge h there and then proceed to node $(2, 1)$,

recharging 0 at $(1,1)$ and h at $(2,1)$ at zero total cost in order to reach the destination (see Figure 3(c)). Otherwise, when the station at node $(0,1)$ is unavailable, the optimal policy is the same as the optimal a priori policy. Then, the cost of the optimal policy with adaptive routing and recharging is $0.5 \cdot (0) + 0.5 \cdot (1) = 0.5$, which is 0.25 less than the cost of the optimal policy with adaptive recharging only and 0.5 less than the cost of the optimal a priori policy. Therefore, both adaptive routing and recharging decisions can lead to a significant reduction in waiting cost.
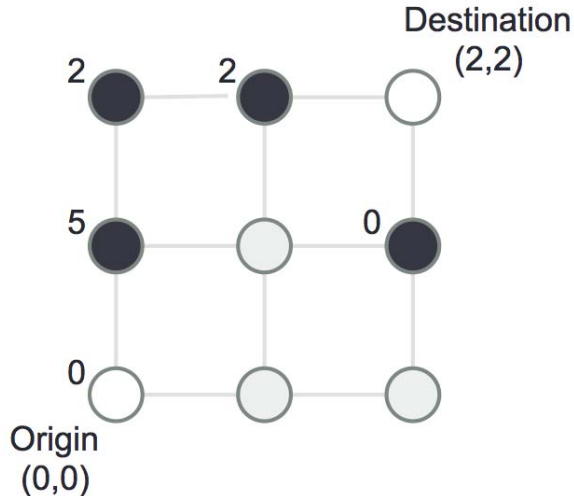

Figure 2: Example network with origin at (0,0) and destination at (2,2). Labels represent $E[W_i|A_i = 0]$, and $P(A_i = 1) = 0.5$ for all black nodes and the origin (gray nodes do not have charging stations, i.e., $P(A_i = 1) = 0$ and $E[W_i|A_i = 0] = \infty$).

3.3 Solution approach

We begin our solution approach by solving for an a priori policy in which the locations of recharging stops and the corresponding recharging amounts at each stop are fixed for the duration of the trip and cannot be changed based on the realized availability of charging stations. We derive efficient solution methods to find optimal a priori policies for a fixed path and for a general grid network setting (Section 4). This a priori policy analysis allows us to establish a benchmark policy for the following adaptive decision making strategies, as well as provide insightful properties beneficial to the adaptive solution approach. In our subsequent analysis of the setting with adaptive decision making, the driver may react to the realized station availability upon arrival at a node. We consider two separate models corresponding to various levels of adaptivity: the EV alters either only its recharging decisions, or both, its path and recharging locations (and quantities) (Section 5). We find an optimal adaptive policy for a fixed path and develop two heuristic approaches for obtaining

(a) Optimal a priori policy

(b) Optimal policy with adaptive recharging

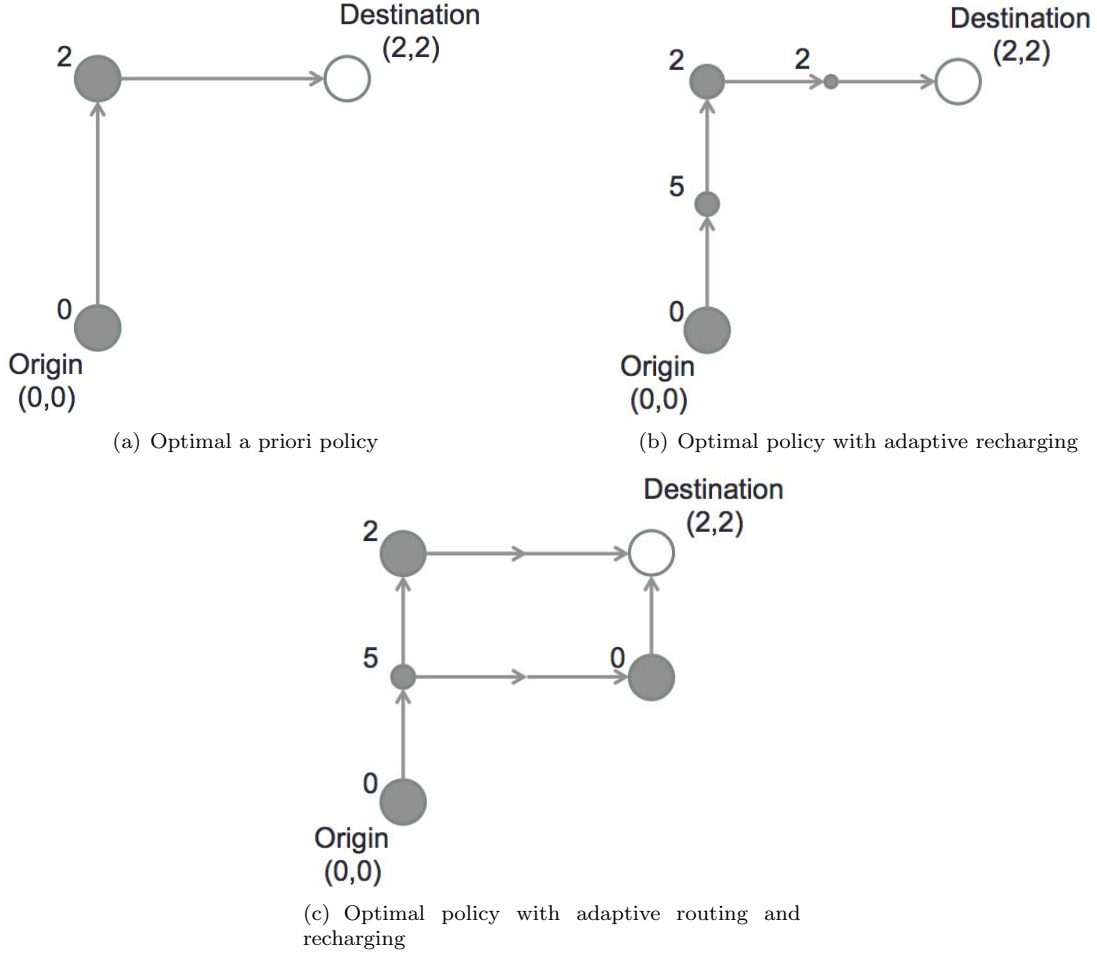(c) Optimal policy with adaptive routing and recharging

Figure 3: Optimal policies with (a) no adaptivity, (b) adaptive recharging only, and (c) adaptive routing and recharging. The size of each node corresponds to the probability that the vehicle recharges at the node under the specified policy given that the vehicle visits the node, with larger sizes representing higher probabilities.

policies in a grid network setting, which utilize the optimal a priori policy as a starting point.

Our analysis of the optimal EV routing is carried out on a square grid network where the amount of charge required to traverse each arc is constant denoted by h (i.e., $h_{ij} = h, \forall (i,j) \in \mathcal{E}$). We consider a grid as opposed to a more general network due to several favorable properties of the square grid network that lend themselves well to a dynamic programming framework. Specifically, consider an x-, y- coordinate system imposed on our grid network, where without loss of generality, we assume that the origin o has coordinates $(0,0)$ and the destination n has coordinates (x, y) $(x \geq 0, y \geq 0)$. Then, the minimum distance between two nodes can be easily calculated as the sum of the absolute differences between their coordinates. Furthermore, network nodes can be partitioned by the total sum of their x- and y- coordinates such that all nodes in one partition have the same sum values. As a result of such procedure, all nodes along each $-\pi/4$

angled diagonal line are grouped together (see Figure 4). Indexing the partitions by the values of the sum of the two coordinates of their member nodes creates a natural ordering. In addition, it can be observed that whenever the vehicle travels to an adjacent node, the index of the diagonal passing through its location either increases by one (when the vehicle travels up or to the right) or decreases by one (when it travels down or to the left). By our convention, the vehicle starts its travel on diagonal 0 and ends on diagonal $x + y$. It follows that for any *direct path* (i.e., shortest path) between origin and destination in which the vehicle only travels either up or to the right, the distance is simply the difference between the partition indices of the two nodes, or $x + y$. This notion of a direct path (as well as *indirect path* – path with distance greater than minimum) will be used in our subsequent analysis.
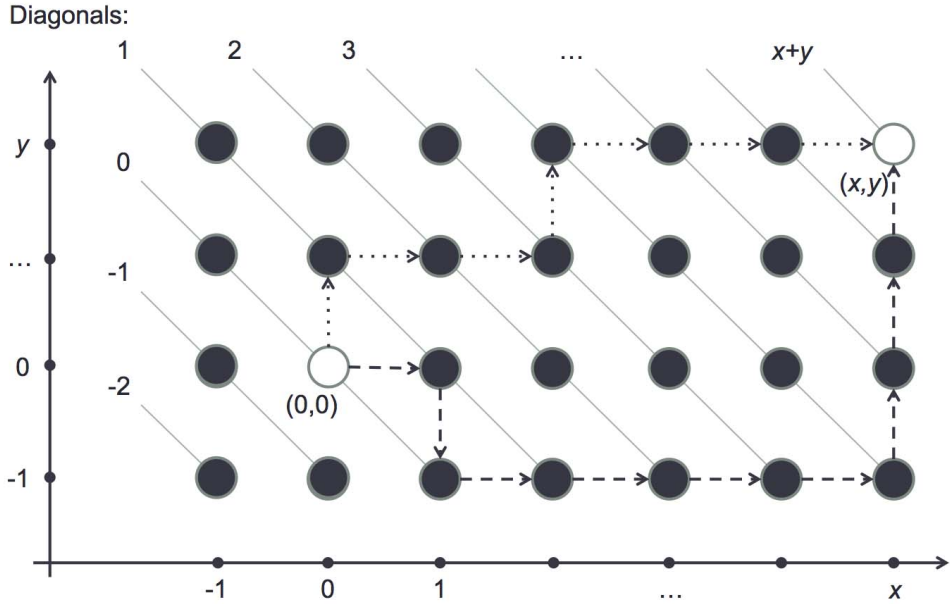

Figure 4: Nodes partitioned diagonally; all nodes along the same diagonal have coordinates that sum to the same value. The dotted line is an example of a *direct path* from the origin to the destination, and the dashed line is an example of an *indirect path*.

4 Optimal a priori policies

Prior to studying an optimal policy with adaptive decision making, it is helpful first to identify an optimal a priori policy in which both the vehicle's route and recharging locations are selected before departure and do not change en route based on the realized availability of charging stations. Not only is an optimal a priori policy easier to obtain, but it is also a feasible policy when adaptive decision making is permitted. As a

result, the value of such a policy serves as an upper bound on the value of an optimal adaptive policy. We apply this fact to our later analysis of the adaptive decision making and use the optimal a priori policy as a starting point for the solution procedure to find an optimal adaptive policy. In this section we present algorithms for finding an optimal a priori policy along a path and within a grid network. In the next section we propose methods for finding adaptive policies using the optimal a priori policy as a benchmark.

4.1   Optimal a priori policy for a fixed path

Here we examine how to determine an optimal a priori recharging policy along a fixed path $\mathcal{N} = (0, ..., n)$ consisting of a sequence of $n + 1$ nodes. To simplify the notation, we let h_i and t_i denote the energy and time, respectively, required to travel from node i to node $i + 1$. We let $\pi_{\mathcal{N}}^0 = (r_0, ..., r_{n-1})$ denote an a priori policy, which consists of the sequence of amounts to recharge at each node along the path $\mathcal{N}$ (except node n, which is the end of the path). Notice that the amounts to recharge at each node are not dependent on the realized availability of charging stations. In such a policy, the expected waiting time at a node if the vehicle recharges there is

$$E[W_i] = [P(A_i = 1)]E[W_i|A_i = 1] + [1 - P(A_i = 1)]E[W_i|A_i = 0] = [1 - P(A_i = 1)]E[W_i|A_i = 0]$$

for all $i \in (1, ..., n)$, where $E[W_i|A_i = 1] = 0$ since the vehicle does not need to wait when the charging station is available. We show in the following lemma that there exists an optimal a priori policy in which the vehicle only stops to recharge when its charge level is zero.

Lemma 1 *There exists an optimal a priori recharging policy such that for all $i \in (0, ..., n-1)$, $r_i > 0$ if and only if $q_i = 0$.*

Proof. Let $\pi_{\mathcal{N}}^0 = (r_0, ..., r_{n-1})$ be an a priori recharging policy. Suppose the vehicle recharges at node j in the policy, and k is the next node along the path where the vehicle recharges (i.e., $k = \min\{\ell : \ell > j, \ r_\ell > 0\}$). Let $\Delta = r_j - \sum_{\ell=j}^{k-1} h_\ell$ denote the amount recharged at node j that is not used by the time the vehicle reaches node k, where $\Delta \geq 0$. Then a policy $\hat{\pi}_{\mathcal{N}}^0 = (\hat{r}_1, ..., \hat{r}_n)$ that satisfies

$$\hat{r}_i = \begin{cases} r_j - \Delta, & i = j \\ r_k + \Delta, & i = k \\ r_i, & \text{otherwise} \end{cases}$$

is also feasible. Furthermore, by the convexity of $c(\cdot)$, the cost of the policy $\hat{\pi}_{\mathcal{N}}^0$ is no greater than that of $\pi_{\mathcal{N}}^0$. Therefore, there exists an optimal policy in which $r_i > 0$ implies $q_i = 0$ for all $i \in (0, ..., n-1)$. Conversely, $q_i = 0$ necessarily implies that $r_i > 0$ in order to ensure feasibility. It follows that there exists an optimal a priori recharging policy such that for all $i \in (0, ..., n-1)$, $r_i > 0$ if and only if $q_i = 0$. ∎

If we let $V(i)$ denote the value function representing the minimum expected cost of reaching node n (the end of the path) when the vehicle arrives at node i with zero charge level, then we can define the value function recursively as

$$V(i) = \min_{j \in (i+1,...,n):\ \sum_{\ell=i}^{j-1} h_\ell \leq q_{max}} \left\{ s + [1 - P(A_i = 1)]E[W_i|A_i = 0] + c\left(\sum_{\ell=i}^{j-1} h_\ell, 0\right) + \sum_{\ell=i}^{j-1} t_\ell + V(j) \right\},$$

where $V(n) = 0$. Solving this recursive equation for $V(0)$ yields an optimal recharging policy for a given path. Note that when the quantity $[1 - P(A_i = 1)]E[W_i|A_i = 0]$ is the same for all $i \in (0, ..., n-1)$, the problem is identical to the one studied in [21] and can be solved more efficiently by removing from the action space values of j for which $\sum_{\ell=i}^{j-1} h_\ell \leq \alpha q_{max}$.

4.2 Optimal a priori policy for a grid network

When searching for an optimal a priori policy in a network, it is important to establish a tight upper bound quickly on its value. This would allow us to significantly reduce the number of network nodes to be considered as part of an optimal path and prevent us from making unnecessary value function updates at nodes that are located far from both the origin and destination. Our solution approach outlined in Algorithm 1 takes advantage of the grid network structure and quickly computes an upper bound on the value of an optimal a priori policy. It then tightens this bound by searching for policies corresponding to longer paths with potential savings in waiting times at the charge stations. The algorithm terminates once it is established that the current upper bound is also a lower bound and the corresponding policy is optimal.

An upper bound on the total cost of an optimal a priori policy is first calculated by the `FindOptDirectPath` procedure (detailed in the Appendix), which finds an optimal policy over all direct paths between origin and destination. Such calculation can be performed efficiently since the vehicle is restricted to traveling either up or to the right toward the destination node, never detouring outside of the bounding box formed by the origin and destination or moving in a direction away from the destination. In addition, the found policy corresponds to a shortest path, since the length of any direct path is $x + y$.

Algorithm 1 Finding an optimal a priori policy in a grid network
---
Input: network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$; destination node (x, y) $(x, y \geq 0)$; max travel distance $k = \left\lfloor \frac{q_{max}}{h} \right\rfloor$
Output: sequence of recharging stops $P^*_{\mathcal{G},(x,y)} = [(i_1^*, j_1^*), ..., (i_p^*, j_p^*)]$ with corresponding cost $V(0,0)$
Initialize: $V(x, y) = 0$; $V(i, j) = \infty$ for every $(i, j) \in \mathcal{N} \backslash (x, y)$; $check(i, j) = 0$ for every $(i, j) \in \mathcal{N}$; $succ(i, j) = (x, y)$ for every $(i, j) \in \mathcal{N}$

1: FindOptDirectPath($\mathcal{G}, x, y, k$)
2: $MinDiag = 0$
3: $MaxDiag = x + y$
4: $ScanType = reverse$
5: **while** $ScanType \neq null$ **do**
6:    **if** $ScanType = reverse$ **then**
7:       ScanReverse($\mathcal{G}, x, y, k, V, check, pred$)
8:    **else if** $ScanType = forward$ **then**
9:       ScanForward($\mathcal{G}, x, y, k, V, check, pred$)
10:    **end if**
11: **end while**
12: $(i, j) = (0, 0)$
13: **while** $(i, j) \neq (x, y)$ **do**
14:    $P^*_{\mathcal{G},(x,y)} = [P^*_{\mathcal{G},(x,y)} \; succ(i, j)]$
15:    $(i, j) = succ(i, j)$
16: **end while**

The algorithm then searches the network for policies with a lower cost by considering paths with longer distance but overall lower cost. It carries this out by alternating between the `ScanReverse` and `ScanForward` procedures until there are no further value function improvements at any of the network nodes. The bound found by `FindOptDirectPath` significantly limits the number of nodes that are explored during each scan step and improves computational efficiency. Starting with the diagonal partition containing the destination node (indexed $x + y$), the `ScanReverse` procedure updates the value function for other nodes along that diagonal, as well as for the nodes on the diagonals with a lower index, yet within a distance of $\lfloor q_{max}/h \rfloor$ from an updated node, k. The procedure then advances to the diagonal with the next highest index, updating nodes along that diagonal, as well as along diagonals with lower indexes, and it continues so until a diagonal along which no nodes have updated value functions is reached. From this diagonal, the `ScanForward` procedure operates analogously to `ScanReverse` but in the forward direction, searching for nodes to update along the same diagonal or along diagonals with a higher index. The final value at the origin upon termination of the algorithm represents the total cost of an optimal a priori policy. (See Appendix for detailed pseudocodes for procedures `FindOptDirectPath`, `ScanReverse` and `ScanForward`.)

The following theorem establishes optimal termination of the algorithm in finite time.

Theorem 2 *Algorithm 1 returns an optimal a priori policy in finite time.*

Proof. The LB function in `ScanReverse` and `ScanForward` procedures of Algorithm 1 (see Appendix) computes the lower bound on the minimum travel time from a given node in the network to the destination node. LB is positive and monotone increasing with respect to the distance between a node and the destination and it restricts the number of nodes to be explored by the algorithm to a finite set. Furthermore, the number of iterations of `ScanReverse` and `ScanForward` is finite because of the suboptimality of cycles. Therefore, Algorithm 1 returns an optimal a priori policy in finite time. ∎

Remark 1 *It is important to note that Algorithm 1 returns an optimal sequence of recharging locations and corresponding recharge amounts, but not an explicit path. Due to the grid network structure, there are always more than one shortest path between any two distinct nodes, as long as they are not horizontally or vertically aligned, and the routing policy corresponding to the found optimal charging policy is non-unique. Therefore, as long as EV driver follows one of many direct paths between each pair of consecutive recharge locations, the resulting routing and recharging policy will be optimal for our a priori setting.*

4.3 Illustrative examples of optimal a priori policies

It is interesting to note that the total number of scan procedures completed (both `ScanReverse` and `ScanForward`) in Algorithm 1 for which the objective value continues to improve directly corresponds to the shape of an optimal path. For example, if an optimal policy is obtained immediately after the `FindOptDirectPath` procedure, then the corresponding path is a direct path between the origin and destination. An improvement to the objective value after one call to the `ScanReverse` procedure implies that the optimal path is not direct but the sequence of diagonals corresponding to each stop is nondecreasing. Additional improvements after the initial `ScanForward` means that the sequence of diagonals has at least one sign change, with further improvements signaling even more sign changes.

Some numerical examples of optimal a priori policies are illustrated in Figure 5. Here, the white area corresponds to nodes with randomly generated nodes availability probability and expected wait times from uniform distributions $U(0, 1)$ and $U(0, 240)$, respectively, while shaded area denotes regions without any charging stations. In the first example (Figure 5(a)), the optimal a priori policy corresponds to a direct path between the origin and destination. This policy is found within the `FindOptDirectPath` procedure, and no improvements to it are made after calls to the `ScanReverse` and `ScanForward` procedures. The second example (Figure 5(b)) illustrates a policy corresponding to an indirect path. Whereas a direct path would have all recharging stops within the bounding box between the origin and destination, the shown path includes stops outside of the bounding box. However, note that the indices of the diagonals corresponding

to the sequence of recharging stops are nondecreasing. This implies that the optimal a priori policy is obtained after the first `ScanReverse` procedure. The third example illustrates an optimal policy that is obtained after multiple calls to the `ScanReverse` and `ScanForward` procedures (Figure 5(c)). In this case, the sequence of diagonals corresponding to the recharging stops has two sign changes (i.e., decreases, then increases and decreases again), implying that the policy was obtained after two `ScanReverse` procedures (and one `ScanForward` procedure).
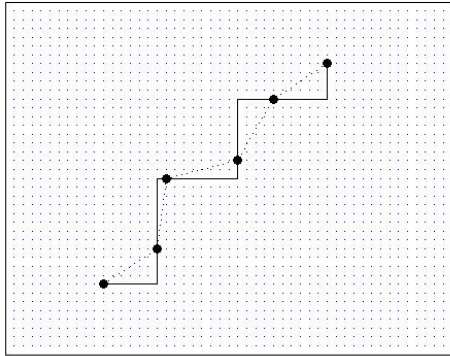
(a) Direct path
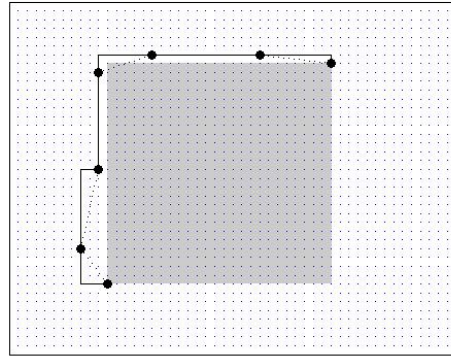
(b) Indirect path (nondecreasing diagonals)
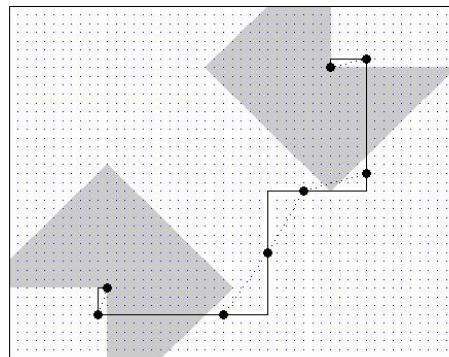
(c) Indirect path (decreasing/increasing diagonals)

Figure 5: Examples of a priori policies in a network. Highlighted nodes represent recharging stops (with a sample path shown as a solid line), and shaded areas represent regions where charging stations are never available.

5 Adaptive decision making

We now consider the problem of finding an optimal routing and recharging policy with adaptive decision making. As opposed to a priori policies in which the path and recharging stops are fixed, adaptive policies allow the driver to make recharging decisions dynamically based on the realized availability of charging stations and also update the route whenever favorable recharging opportunities are encountered. We begin by finding an optimal adaptive policy for a given path and then extend the analysis to the grid network setting. We consider two distinct models with different levels of adaptive decision making allowed to the drive: (1) the path is fixed for the duration of the trip and only adaptive recharging decisions are considered, and (2) both, routing and recharging, decisions are made by the driver adaptively along the way. As a result, two heuristic solution methods are developed for finding adaptive policies in a grid network.

5.1 Adaptive recharging policy for a selected path

In this section, we solve for an optimal recharging policy along a fixed path $\mathcal{N} = (0, ..., n)$ consisting of a sequence of $n+1$ nodes. As in Section 4.1, we let h_i and t_i denote the energy and time, respectively, required to travel from node i to node $i + 1$. Due to the fact that there are no routing decisions to be made and the vehicle always travels forward along the path, the choice of which node to visit next is fixed and thus the action space reduces to

$$\mathcal{A}_{(i,q_i)} = \{(j, r_i) : j = i + 1, \ r_i \in [(h_i - q_i)^+, q_{max} - q_i]\}.$$

If we instead embed the fixed path in the value function representation from (3) and use

$$\mathcal{A}^R_{(i,q_i)} = \{r_i : r_i \in [(h_i - q_i)^+, q_{max} - q_i]\}$$

to denote the part of the action space consisting only of the feasible recharging amounts at each node, then we can rewrite the value function as

$$V(i, q_i, a_i) = \min_{r_i \in \mathcal{A}^R_{(i,q_i)}} \{t_i + (s + (1 - a_i)E[W_i|A_i = 0])I_{r_i>0} + c(r_i, q_i) \ +$$

$$[P(A_{i+1} = 1)]V(i + 1, q_i + r_i - h_i, 1) + [1 - P(A_{i+1} = 1)]V(i + 1, q_i + r_i - h_i, 0)\}.$$

To denote a feasible recharging policy for the path, we use $\pi^R_{\mathcal{N}} : \mathcal{S} \to \mathcal{A}^R$, with $\pi^R_{\mathcal{N}}(i, q_i, a_i) \in \mathcal{A}^R_{(i,q_i)}$

representing the recharging amount specified by the policy $\pi_{\mathcal{N}}^R$ for the state (i, q_i, a_i). The expected cost of reaching the end of the path under such a policy for a given state (i, q_i, a_i), which we denote $C_{\pi_{\mathcal{N}}^R}(i, q_i, a_i)$, satisfies

$$C_{\pi_{\mathcal{N}}^R}(i, q_i, a_i) = t_i + (s + (1 - a_i)E[W_i|A_i = 0])I_{\pi_{\mathcal{N}}^R(i,q_i,a_i)>0} + c(\pi_{\mathcal{N}}^R(i, q_i, a_i), q_i) +$$

$$[P(A_{i+1} = 1)]C_{\pi_{\mathcal{N}}^R}(i + 1, q_i + \pi_{\mathcal{N}}^R(i, q_i, a_i) - h_i, 1) +$$

$$[1 - P(A_{i+1} = 1)]C_{\pi_{\mathcal{N}}^R}(i + 1, q_i + \pi_{\mathcal{N}}^R(i, q_i, a_i) - h_i, 0),$$

where $I_{\pi_{\mathcal{N}}^R(i,q_i,a_i)>0}$ equals 1 if $\pi_{\mathcal{N}}^R(i, q_i, a_i) > 0$ and 0 otherwise. We let $\Pi_{\mathcal{N}}^R$ denote the set of all feasible recharging policies for the path $\mathcal{N}$, and thus the value function can be expressed in terms of the cost function $C_{\pi_{\mathcal{N}}^R}(\cdot)$ as

$$V(i, q_i, a_i) = \min_{\pi_{\mathcal{N}}^R \in \Pi_{\mathcal{N}}^R} C_{\pi_{\mathcal{N}}^R}(i, q_i, a_i).$$

Finding an optimal recharging policy is not practical when the vehicle's charge level can take on any value in the interval $[0, q_{max}]$ since this corresponds to a continuous action space and state space. However, without loss of optimality, we can restrict the possible values for the recharging amounts at each node to a discrete set. Then, we consider only finitely many values for the vehicle's charge level, thereby making the problem of finding an optimal recharging policy tractable. The following lemma and corollary show that there exists an optimal recharging policy such that all of the recharging amounts (and also incoming charge levels at each node) can be expressed as sums of the h_i values.

Lemma 3 *There exists an optimal recharging policy $\pi_{\mathcal{N}}^{R\,*} : \mathcal{S} \to \mathcal{A}^R$ such that*

$$\pi_{\mathcal{N}}^{R\,*}(i, q_i, a_i) \in \left\{ \left(\sum_{\ell=i}^{k-1} h_\ell - q_i \right)^+ : k \in (i + 1, ..., n - 1), \sum_{\ell=i}^{k-1} h_\ell \le q_{max} \right\}$$

for all $(i, q_i, a_i) \in \mathcal{S}$.

Proof. Suppose $\pi_{\mathcal{N}}^R : \mathcal{S} \to \mathcal{A}^R$ is a feasible recharging policy, and there exists a state $(i, q_i, a_i) \in \mathcal{S}$ such that $\pi_{\mathcal{N}}^R(i, q_i, a_i) = \sum_{\ell=i}^{k-1} h_\ell - q_i + \Delta > 0$ for some $k \in (i + 1, ..., n - 1)$ and $0 \le \Delta < h_k$. Then consider an

alternate policy $\pi_{\mathcal{N}}^{R\,'} : \mathcal{S} \to \mathcal{A}^R$ where

$$\pi_{\mathcal{N}}^{R\,'}(j, q_j, a_j) = \begin{cases} \pi_{\mathcal{N}}^R(j, q_j, a_j) - \Delta, & (j, q_j, a_j) = (i, q_i, a_i) \\ \pi_{\mathcal{N}}^R(j, q_j + \Delta, a_j) + \Delta I_{\pi_{\mathcal{N}}^R(j, q_j + \Delta, a_j) > 0}, & j \in (i+1, ..., k) \text{ and } q_j = \sum_{\ell=j}^{k-1} h_\ell \\ \pi_{\mathcal{N}}^R(j, q_j, a_j), & \text{otherwise} \end{cases}$$

for any $(j, q_j, a_j) \in \mathcal{S}$, where $I_{\pi_{\mathcal{N}}^R(j, q_j + \Delta, a_j) > 0}$ equals 1 if $\pi_{\mathcal{N}}^R(j, q_j + \Delta, a_j) > 0$ and 0 otherwise. In this alternate policy, the vehicle's charge level when departing from node i is $\sum_{\ell=i}^{k-1} h_\ell$ instead of $\sum_{\ell=i}^{k-1} h_\ell + \Delta$. The next node where the vehicle stops to recharge is the same as in $\pi_{\mathcal{N}}^R$, although the vehicle recharges an extra amount Δ so that the resulting charge level is the same as in the original policy. Beyond that node, both policies are identical.

Suppose that j is the next node after i at which the vehicle stops to recharge for a given sequence $(a_{i+1}, ..., a_j)$ of observed station availabilities (i.e., $A_\ell = a_\ell$ for all $\ell \in (i+1, ..., j)$). Then the expected cost of the policy $\pi_{\mathcal{N}}^R$ for the state (i, q_i, a_i) is

$$C_{\pi_{\mathcal{N}}^R}(i, q_i, a_i) \mid (A_\ell = a_\ell \; \forall \; \ell \in (i+1, ..., j)) = \sum_{\ell=i}^{j-1} t_\ell + s + (1 - a_i) E[W_i | A_i = 0] + c \left(\sum_{\ell=i}^{k-1} h_\ell - q_i + \Delta, q_i \right) +$$

$$C_{\pi_{\mathcal{N}}^R} \left(j, \sum_{\ell=j}^{k-1} h_\ell + \Delta, a_j \right). \tag{4}$$

Now suppose that the same sequence of station availabilities is observed under policy $\pi_{\mathcal{N}}^{R\,'}$. Then the expected cost of the policy under such a scenario is

$$C_{\pi_{\mathcal{N}}^{R\,'}}(i, q_i, a_i) \mid (A_\ell = a_\ell \; \forall \; \ell \in (i+1, ..., j)) = \sum_{\ell=i}^{j-1} t_\ell + s + (1 - a_i) E[W_i | A_i = 0] + c \left(\sum_{\ell=i}^{k-1} h_\ell - q_i, q_i \right) +$$

$$C_{\pi_{\mathcal{N}}^{R\,'}} \left(j, \sum_{\ell=j}^{k-1} h_\ell, a_j \right). \tag{5}$$

When we compare the costs of the two policies for the given realization of observed station availabilities, we find that

$$(4) - (5) = c \left(\sum_{\ell=i}^{k-1} h_\ell - q_i + \Delta, q_i \right) + C_{\pi_{\mathcal{N}}^R} \left(j, \sum_{\ell=j}^{k-1} h_\ell + \Delta, a_j \right) - c \left(\sum_{\ell=i}^{k-1} h_\ell - q_i, q_i \right) - C_{\pi_{\mathcal{N}}^{R\,'}} \left(j, \sum_{\ell=j}^{k-1} h_\ell, a_j \right)$$

$$= c\left(\Delta, \sum_{\ell=i}^{k-1} h_\ell\right) + C_{\pi_{\mathcal{N}}^R}\left(j, \sum_{\ell=j}^{k-1} h_\ell + \Delta, a_j\right) - C_{\pi_{\mathcal{N}}^{R\prime}}\left(j, \sum_{\ell=j}^{k-1} h_\ell, a_j\right)$$

$$= c\left(\Delta, \sum_{\ell=i}^{k-1} h_\ell\right) + c\left(\pi_{\mathcal{N}}^R\left(j, \sum_{\ell=j}^{k-1} h_\ell + \Delta, a_j\right), \sum_{\ell=j}^{k-1} h_\ell + \Delta\right) - c\left(\pi_{\mathcal{N}}^{R\prime}\left(j, \sum_{\ell=j}^{k-1} h_\ell, a_j\right), \sum_{\ell=j}^{k-1} h_\ell\right)$$

$$= c\left(\Delta, \sum_{\ell=i}^{k-1} h_\ell\right) + c\left(\pi_{\mathcal{N}}^R\left(j, \sum_{\ell=j}^{k-1} h_\ell + \Delta, a_j\right), \sum_{\ell=j}^{k-1} h_\ell + \Delta\right) - c\left(\pi_{\mathcal{N}}^R\left(j, \sum_{\ell=j}^{k-1} h_\ell + \Delta, a_j\right) + \Delta, \sum_{\ell=j}^{k-1} h_\ell\right)$$

$$= c\left(\Delta, \sum_{\ell=i}^{k-1} h_\ell\right) - c\left(\Delta, \sum_{\ell=j}^{k-1} h_\ell\right)$$

$$= \left(\gamma\Delta + f\left(\left(\sum_{\ell=i}^{k-1} h_\ell + \Delta - \alpha q_{max}\right)^+\right) - f\left(\left(\sum_{\ell=i}^{k-1} h_\ell - \alpha q_{max}\right)^+\right)\right) -$$

$$\left(\gamma\Delta + f\left(\left(\sum_{\ell=j}^{k-1} h_\ell + \Delta - \alpha q_{max}\right)^+\right) - f\left(\left(\sum_{\ell=j}^{k-1} h_\ell - \alpha q_{max}\right)^+\right)\right)$$

$$\geq \nabla f\left(\left(\sum_{\ell=i}^{k-1} h_\ell - \alpha q_{max}\right)^+\right)\Delta - \nabla f\left(\left(\sum_{\ell=j}^{k-1} h_\ell + \Delta - \alpha q_{max}\right)^+\right)\Delta \quad \text{(by convexity of } f(\cdot))$$

$$\geq 0 \tag{6}$$

since $\sum_{\ell=i}^{k-1} h_\ell > \sum_{\ell=j}^{k-1} h_\ell + \Delta$. The expression in (6) holds for any given sequence $(a_{i+1}, ..., a_j)$ of observed station availabilities such that j is the next node after i where the vehicle recharges, and because the probability of the sequence occurring is the same under both policies, the (unconditional) expected cost of the policy $\pi_{\mathcal{N}}^{R\prime}$ is no greater than that of $\pi_{\mathcal{N}}^R$ for the state (i, q_i, a_i). It follows that there exists an optimal recharging policy $\pi_{\mathcal{N}}^{R*} : \mathcal{S} \to \mathcal{A}^R$ such that

$$\pi_{\mathcal{N}}^{R*}(i, q_i, a_i) \in \left\{\left(\sum_{\ell=i}^{k-1} h_\ell - q_i\right)^+ : k \in (i+1, ..., n-1), \sum_{\ell=i}^{k-1} h_\ell \leq q_{max}\right\}$$

for all $(i, q_i, a_i) \in \mathcal{S}$. ∎

Since we assume that the vehicle's charge level is zero at the beginning of the path, it follows from Lemma 3 that there exists an optimal recharging policy such that every incoming charge level and recharging amount can be expressed as a summation of the h_ℓ parameters. We prove this claim in the following corollary.

Corollary 4 *Let*

$$\mathcal{H}(i) = \left\{ \sum_{\ell=i}^{k-1} h_\ell : k \in (i, ..., n-1), \ \sum_{\ell=i}^{k-1} h_\ell \leq q_{max} \right\}$$

denote the set of feasible charge levels at node i that can be expressed as a summation of the h_ℓ parameters. Then there exists an optimal recharging policy $\pi_{\mathcal{N}}^{R\,*} : \mathcal{S} \to \mathcal{A}^R$ such that $q_i = \sum_{\ell=i}^{k-1} h_\ell \in \mathcal{H}(i)$ for some $k \in (i, ..., n-1)$ and $\pi_{\mathcal{N}}^{R\,*}(i, q_i, a_i) \in \mathcal{H}(k)$ for any realized state $(i, q_i, a_i) \in \mathcal{S}$, given that the initial state is $(0, 0, \cdot)$.

Proof. We prove the claim by induction. At node 0, we have $q_0 = 0 \in \mathcal{H}(1)$, and by Lemma 3, there exists an optimal recharging policy $\pi_{\mathcal{N}}^{R\,*}$ such that $\pi_{\mathcal{N}}^{R\,*}(0, 0, \cdot) \in \mathcal{H}(1)$.

Then assuming the claim holds for a node $i \in (0, ..., n-2)$, suppose $q_i = \sum_{\ell=i}^{j-1} h_\ell \in \mathcal{H}(i)$ for some $j \in (i, ..., n-1)$ and $\pi_{\mathcal{N}}^{R\,*}(i, q_i, a_i) = \sum_{\ell=j}^{j'-1} h_\ell \in \mathcal{H}(j)$ for some $j' \in (j, ..., n-1)$. At node $i+1$, the vehicle's charge level is

$$q_{i+1} = \sum_{\ell=i}^{j-1} h_\ell + \sum_{\ell=j}^{j'-1} h_\ell - h_i = \sum_{\ell=i+1}^{j'-1} h_\ell \in \mathcal{H}(i+1),$$

and by Lemma 3, we have

$$\pi_{\mathcal{N}}^{R\,*}(i+1, q_{i+1}, \cdot) \in \left\{ \left(\sum_{\ell=i+1}^{k-1} h_\ell - q_{i+1} \right)^+ : \ k \in (i+2, ..., n-1), \ \sum_{\ell=i}^{k-1} h_\ell \leq q_{max} \right\}$$

$$\subseteq \left\{ \sum_{\ell=j'}^{k-1} h_\ell : \ k \in (j', ..., n-1), \ \sum_{\ell=i}^{k-1} h_\ell \leq q_{max} \right\}$$

$$\subseteq H(j').$$

Thus, the claim holds for node $i+1$ as well, and it follows that $q_i \in \mathcal{H}(i)$ and $\pi_{\mathcal{N}}^{R\,*}(i, q_i, a_i) \in \mathcal{H}(k)$ for any realized state $(i, q_i, a_i) \in \mathcal{S}$ given that the initial state is $(0, 0, \cdot)$. ∎

By Corollary 4, without loss of optimality, we can create a value function representation that considers only a discrete set of values for both the charge level and recharging amount at each node. If we let $V(i, k, a_i)$ denote the value function representing the minimum expected cost of reaching node n when we arrive at node i with charge level $\sum_{\ell=i}^{k-1} h_\ell$ (i.e., just enough to reach node k) and the station availability is a_i, then we can define the value function recursively as

$$V(i, k, a_i) = \min_{j \in (k, ..., n): \ \sum_{\ell=i}^{j-1} h_\ell \leq q_{max}} \left\{ t_i + (s + (1 - a_i)E[W_i | A_i = 0]) I_{j>k} + c \left(\sum_{\ell=k}^{j-1} h_\ell, \sum_{\ell=i}^{k-1} h_\ell \right) + \right.$$

$$[P(A_{i+1} = 1)]V(i+1, j, 1) + [1 - P(A_{i+1} = 1)]V(i+1, j, 0) \bigg\}, \quad (7)$$

where $I_{j>k}$ equals 1 if $j > k$ (i.e., if the vehicle stops to recharge) and 0 otherwise, and $\sum_{\ell=k}^{j-1} h_\ell$ is the recharging amount. Then, solving the recursive equation (7) until we find $V(0, 0, \cdot)$ will result in an optimal recharge policy along the given path.

5.2 Adaptive policies for a grid network

We focus our attention in this section on finding adaptive policies in a grid network. (Recall motivating example in Section 3.2.) Due to computational demands described below, we develop heuristic solution approaches to estimate the value of adaptive decision making and use an optimal a priori policy to benchmark the improvement of EV performance resulting from introducing adaptivity. We first compute the benefit of adaptive recharging decisions only, and then we determine the additional savings that result from adaptive routing decisions and recharging strategies.

Although solving the dynamic programming formulation in (7) to obtain an optimal recharging policy along a fixed path is straightforward, applying a similar solution approach to find an optimal policy within a network is more difficult due to the need to incorporate routing decisions as well. Thus, the decision space increases to include the next node to visit on the EV route and the set of states to be considered increases to include potentially all nodes of the road network. Furthermore, since the order of the visited nodes is not prespecified as it is done in a fixed path setting, value function updates could require re-evaluation of previously explored paths and policies. At the same time, in comparison to an optimal a priori solution in a network, the routing decisions between the points of recharge are integrated into the adaptive decision making since it impacts the nodes EV visits as potential recharge locations if they happen to be available. These facts lead to significant computational inefficiencies unless the number of calculations can be reduced. Therefore, rather than attempt to solve for an optimal policy directly we develop heuristic approaches motivated by our earlier analysis and findings.

We now describe two heuristic methods for obtaining adaptive policies in a grid network. The first generates multiple sample paths to quickly estimate the value of adaptive recharging decisions and selects the best path, while the second heuristic captures additional adaptivity with regard to path selection in order to provide further improvement.

5.2.1 Heuristic 1: Adaptive recharging only

Our first heuristic is broken down into a two-stage solution approach: at the first stage we select what path EV should follow from origin to destination, and then the second stage selects an adaptive recharging policy for the selected path. As shown in Section 5.1, an optimal recharging policy along a fixed path can be obtained using the formulation (7). We use that method to "price" each potential path considered by the first-stage problem. Now, to bound the set of paths considered by the algorithm, we use the optimal a priori policy found by Algorithm 1. Recall that an optimal a priori policy in a network does not always uniquely specify a path but only identifies the optimal recharging locations to be used by the EV (see Remark 1). Therefore, in our first heuristic, outlined in Algorithm 2, we construct $numPaths$ different randomly sampled paths that contain the nodes corresponding to the recharging stops in the optimal a priori policy. With the aim to minimize travel and recharging costs, we require that our sampled paths be direct paths between each consecutive pair of a priori stop locations with no detours or cycles. We then calculate the value of an optimal adaptive policy along each path and choose the path with the lowest value. Subsequently, we compare the value of this found policy with the optimal a priori solution value and use the difference between these values as an estimation for the benefit of adaptive decisions making (restricted to recharging only).

Algorithm 2 Improving optimal a priori policies in a grid network with adaptive recharging decisions

---

Input: network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$; $P(A_{(i,j)} = 1)$ and $E[W_{(i,j)}|A_{(i,j)} = 0]$ for all $(i,j) \in \mathcal{N}$; origin node (0,0) and
    destination node (x,y) $(x, y \geq 0)$; max travel distance $k = \left\lfloor \frac{q_{max}}{h} \right\rfloor$; optimal a priori recharging stops
    $P^*_{\mathcal{G},(x,y)} = [(i^*_1, j^*_1), ..., (i^*_p, j^*_p)]$ with corresponding cost V^*_0; number of paths to construct $numPaths$
Output: value of adaptive recharging policy v^*
Initialize: $V(x, y, \cdot, \cdot) = 0$; $V(i, j, q, a) = \infty$ for all $(i,j) \in \mathcal{N}\backslash(x,y)$, $q \in \{0, h, ..., (k-1)h\}$, and $a \in \{0, 1\}$

1: set $v^* = V^*_0$
2: **for** $path = 1, ..., numPaths$ **do**
3:     construct path $P = [(0,0), ..., (x,y)]$ between $(0,0)$ and (x,y) that contains all nodes in $P^*_{\mathcal{G},(x,y)}$ and is
4:      a direct path between each pair of consecutive nodes in $P^*_{\mathcal{G},(x,y)}$
5:     **for** $a = 0, 1$ **do**
6:       evaluate $V(0,0,0,a)$ for path P using equation (7)
7:     **end for**
8:     set $v^* = \min\left\{v^*, E_{A_{(0,0)}}[V(0,0,0,A_{(0,0)})]\right\}$
9: **end for**

Because we choose $numPaths$ to be less than the value that would be required for total enumeration over all possible paths, we are not guaranteed to find the path with the lowest optimal value. The runtime of the algorithm increases linearly with the number of paths explored. At the same time, the expected marginal improvement that each additional path delivers diminishes fairly rapidly - a fact supported by our numerical results. The goal of this heuristic is to generate a quick and reasonable policy with adaptive recharging

decisions. Our next heuristic incorporates adaptive path selection to provide an improved adaptive policy with a lower cost.

5.2.2 Heuristic 2: Adaptive routing and recharging

In our second heuristic, outlined in Algorithm 3, we include both adaptive routing and recharging decisions. As with Heuristic 1, we require that the vehicle visit (but not necessarily recharge at) every stop location in the optimal a priori policy and travel along a direct path between consecutive a priori stop locations. Rather than fix the entire path prior to solving for the optimal policy, however, we allow the vehicle to adjust its trajectory as it travels from the origin $(0,0)$ to the destination (x,y). Working backwards from the destination (x,y), Algorithm 3 finds an optimal policy between consecutive a priori stop locations over all possible states (i.e., not just those for which the vehicle's charge level at a stop location is 0). It considers all direct paths between each pair of nodes and permits adaptive decision making for both routing and recharging. Because this method takes advantage of the grid structure as it searches for an optimal policy, it is very efficient and terminates quicker than Algorithm 2 in many cases, even when $numPaths$ is small.

Algorithm 3 Improving optimal a priori policies in a grid network with adaptive routing and recharging decisions

---

Input: network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$; $P(A_{(i,j)} = 1)$ and $E[W_{(i,j)}|A_{(i,j)} = 0]$ for all $(i,j) \in \mathcal{N}$; origin node $(0,0)$ and destination node (x,y) $(x,y \geq 0)$; max travel distance $k = \left\lfloor \frac{q_{max}}{h} \right\rfloor$; optimal a priori recharging stops $P^*_{\mathcal{G},(x,y)} = [(i_1^*, j_1^*), ..., (i_p^*, j_p^*)]$ with corresponding cost V_0^*; maximum consecutive a priori stops to skip $maxSkip$

Output: value of adaptive routing and recharging policy v^*

Initialize: $V(x,y,\cdot,\cdot) = 0$; $V(i,j,q,a) = \infty$ for all $(i,j) \in \mathcal{N} \backslash (x,y)$, $q \in \{0, h, ..., (k-1)h\}$, and $a \in \{0,1\}$

1: define $(i_{p+1}^*, j_{p+1}^*) = (x,y)$
2: **for** $skip = 0, ..., maxSkip$ **do**
3:    **for** $\ell = p - skip, p - skip - 1, ..., 1$ **do**
4:       define $\mathcal{G}_\ell$ as the subnetwork consisting of all nodes and edges that are part of a direct path between
5:       (i_ℓ^*, j_ℓ^*) and $(i_{\ell+skip}^*, j_{\ell+skip}^*)$
6:       **for** $q = 0, h, ..., (k-1)h$ and $a = 0, 1$ **do**
7:         evaluate $V(i_\ell^*, j_\ell^*, q, a)$ for subnetwork $\mathcal{G}_\ell$ using equation (3)
8:       **end for**
9:    **end for**
10: **end for**
11: set $v^* = E_{A_{(0,0)}}[V(i_1^*, j_1^*, 0, A_{(0,0)})]$

To relax the restriction that every stop location in the optimal a priori policy be visited, the parameter $maxSkip$ can be increased from 0 up to a maximum of $p - 1$ (where p is the number of stops in the optimal a priori policy). The value of $maxSkip$ corresponds to the maximum number of consecutive a priori stop locations that may be skipped in order to find an adaptive routing and recharging policy with an even lower

cost. For example, when $maxSkip = 0$, every stop location in the optimal a priori policy must be visited (although the vehicle need not stop to recharge at each location). In the case when $maxSkip = 2$, a policy may skip one or two consecutive a priori stop locations at a time, provided that the vehicle follows a direct path between the a priori stop locations it does visit. Then, when $maxSkip = p - 1$, any policy in which the vehicle travels along a direct path between the origin $(0, 0)$ and destination (x, y) is permissible. Although the motivating example discussed in Section 3.2 suggests that increasing $maxSkip$ can significantly lower the total cost, our numerical results indicate that in most cases the improvement is negligible and not worth the significant increase in computational time. Therefore, in the next section, we fix $maxSkip = 0$ so that only policies that include all a priori stop locations are compared.

6 Numerical results

We implemented Algorithms 1, 2, and 3 for a variety of simulated urban settings in order to compare their performance. We use as our network a 500-by-500 node grid with a distance of five miles between adjacent nodes. Expected waiting times and station availabilities at each node are generated from separate uniform distributions as given in Table 1. In Scenario 1, charging stations are available on average half of the time and have expected wait times (if occupied) of around two hours, whereas in Scenario 2, charging stations are more likely to be available but also have longer expected wait times. Thus, the average unconditional expected wait time for a station is similar under each scenario, but the variances are different. For each scenario, we consider a set of 50 random origin-destination pairs separated by distances of between 100 and 300 miles representing trips that an EV driver could reasonably complete in a day, and also another set of 50 random origin-destination pairs with distances ranging from 500 to 1,000 miles that require multiple recharging stops.

	Distributions	
Scenario	$E[W_{i,j}|A_{i,j} = 0]$	$P(A_{i,j} = 1)$
1	$U(0, 240)$	$U(0, 1)$
2	$U(240, 480)$	$U(0.8, 1)$

Table 1: Distributions of expected wait times (measured in minutes) and availability probabilities at each charging station

We test each algorithm over every scenario and set of trips and for all possible combinations of the parameter settings listed in Table 2. We fix the value for α as 0.8 since that is the most commonly used value in practice [14], and the range of values for q_{max} corresponds to the maximum driving range of the

2014 Nissan Leaf under different traffic and environmental conditions [5]. For the overcharging cost function, we use

$$f(z) = 2 \cdot (\exp\{z/5\} - 1), \tag{8}$$

where z is the amount overcharged at a given stop. The expression in (8) captures the real-world dynamics of battery charging and is calibrated based on one driver's experience with overcharging a Leaf [9].

Parameter	Value(s)	Units
α	0.8	N/A
γ	1	time
h	5	energy
q_{max}	$\{70, 80, 90, 100\}$	energy
s	$\{1, 10, 100\}$	time
t	$\{5, 10, 15\}$	time
$numPaths$	5	N/A
$maxSkip$	0	N/A

Table 2: Model parameter settings

6.1 Value of adaptive decision making

In this section, we measure the value of adaptive decision making by comparing the costs of the policies obtained by the two heuristic methods with that of an optimal a priori policy. Note that the primary benefit of an adaptive policy is a reduction in the total expected waiting time for a given trip. Therefore, rather than compare the aggregate cost savings obtained from Algorithms 2 and 3 to the total cost of a policy found by Algorithm 1, we instead compare them only to the total expected waiting cost of an optimal a priori policy. This provides a better reference for understanding the potential savings that adaptivity allows.

The value of adaptive decision making in the two heuristics is illustrated in Figure 6 for trips of different lengths and for both grid scenarios. Because Heuristic 2 allows for greater adaptivity than Heuristic 1, it always provides a higher benefit. The benefit is not always so pronounced, however, since the system is ergodic and the vehicle still must visit all of the a priori stop locations (because $maxSkip = 0$), and thus the ability to explore alternative paths is limited. In addition, the value of adaptivity is greater for longer trips and for Scenario 2. Long trips have more opportunities for the vehicle to find available charging stations without having to wait, and there is also greater flexibility with regard to path selection due to the higher number of nodes between the origin and destination. Comparing the two grid scenarios, the value of adaptivity is higher in Scenario 2 because the expected wait times at unavailable charging stations are

27

longer than in Scenario 1 despite the unconditional expected wait times ($E[W_i]$) being similar. The vehicle is more likely to encounter an available charging station before running out of charge and being forced to stop at a station regardless of its availability, whereas in Scenario 1 there is less incentive to recharge earlier than necessary.
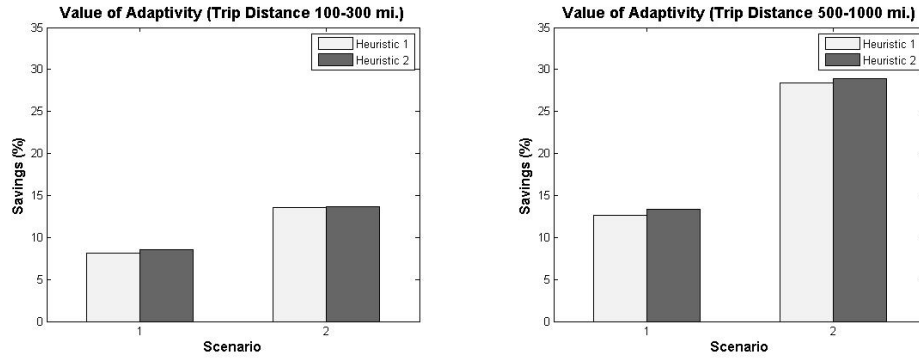

Figure 6: Value of adaptive decision making as a percentage of the total expected waiting time in an optimal a priori policy

6.2 Number of stops comparison

As a result of the vehicle being able to make adaptive recharging decisions and take advantage of opportunities to recharge at available charging stations, the stopping locations tend to occur sooner along the route than in the optimal a priori policy. When this effect is compounded over the length of the entire path, it can lead to instances in which the number of stops in an adaptive policy is greater than the number of stops in an a priori policy. Figure 7 shows the additional number of stops in the heuristic policies relative to that of the optimal a priori policy. Understandably, the longer trips tend to have a greater number of stops when adaptive decisions are permitted since the combined effect of recharging earlier than necessary is greater and is more likely to lead to an extra stop.

Averaged over all instances, the net additional increase in the number of stops is less than 2%. However, it can be seen that the effect is greater for Scenario 1 than for Scenario 2, which implies that the adaptive policies are more likely to stop to recharge if a charging station is available. This is due to the fact that the average probability of a charging station being available is lower in Scenario 1 than in Scenario 2, and thus the vehicle is more likely to charge sooner than necessary to avoid the chance of having to stop at an unavailable station. Heuristic 1 also tends to have more stops than Heuristic 2 because when the path is fixed, there are fewer stations from which to choose in order to delay recharging. With adaptive routing, the
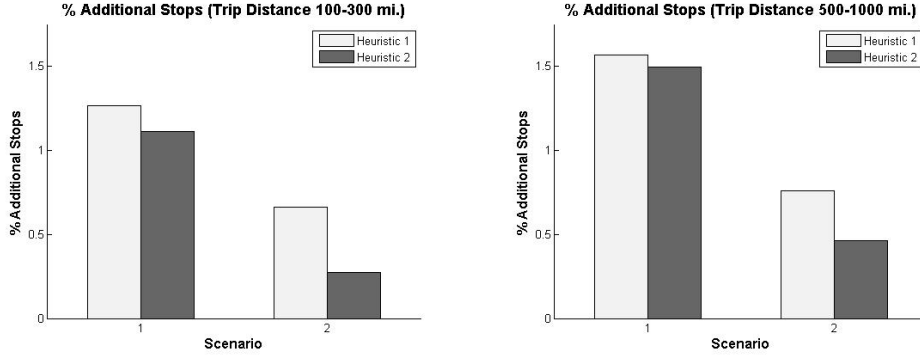
Figure 7: Average difference in number of stops between heuristic and optimal a priori policies

vehicle has several options and can pick the path that offers the greatest opportunity to delay recharging and reduce the likelihood of having to make an additional stop later along its route. Such opportunities are more prevalent in Scenario 2 than in Scenario 1, as the mean station availability is higher, and thus the difference in the number of stops between the two heuristics is greater in Scenario 2.

7 Conclusion

In this paper, we explore the problem of finding an optimal routing and recharging policy in a grid network with uncertain charging station availability. We begin by identifying properties of optimal policies, and using these properties, we develop efficient algorithms for finding an optimal a priori policy for a path and for a grid network. We then introduce adaptive routing and recharging decisions and show how to find an optimal adaptive policy for a fixed path. We also present and implement two heuristics for obtaining an adaptive policy for a grid network, and we conduct detailed numerical experiments to evaluate the performance of our solution methods and estimate the value of adaptivity.

The uncertainty of charging station availability within the network and the ability of the vehicle to adaptively choose which route to follow and where to recharge are unique and important features of our model. Our inclusion of a convex recharging cost function further improves the suitability of this work for capturing the costs incurred by EV drivers. As long as charging stations remain scarce and range anxiety persists, it is critical for EV drivers to be equipped with the necessary decision-support tools to minimize their time spent waiting for charging stations to become available and also make optimal routing and recharging decisions.

As part of our future work, we plan to further explore the value of adaptivity. We will refine our solution

methods to identify additional opportunities within a network setting for an EV driver to benefit from adaptive decision making. We also intend to determine the value of an EV driver knowing not only the availability of the charging station at his or her current location, but also the availability of other nearby charging stations. Having such online information could potentially be of great benefit by enabling the driver to anticipate the likelihood of distant charging stations being available and plan his or her route accordingly. However, the exponential increase in computational time required to incorporate distant stations into a driver's decisions would need to be addressed.

Appendix

7.1 Code for procedures used in Algorithm 1

```
 1: procedure FindOptDirectPath(𝒢, x, y, k)
 2: {
 3: G = {(i, j) ∈ 𝒩 : 0 ≤ i ≤ x,  0 ≤ j ≤ y}
 4: check(0, 0) = 1
 5: for diag = 0, ..., x + y − 1 do
 6:    for (i, j) ∈ G : i + j = diag do
 7:       if V(i, j) + LB(|x − i| + |y − j|) < V(x, y) then
 8:          for all (Δi, Δj) : Δi ≥ 0,  Δj ≥ 0,  0 < Δi + Δj < k,  (i + Δi, j + Δj) ∈ G do
 9:             v = V(i, j) + (1 − E[A_i])E[W_i|A_i = 0] + s + c((Δi + Δj) · h, 0) + (Δi + Δj) · t
10:             if v < V(i + Δi, j + Δj) then
11:                V(i + Δi, j + Δj) = v
12:                pred(i + Δi, j + Δj) = (i, j)
13:                check(i + Δi, j + Δj) = 2
14:             end if
15:          end for
16:       end if
17:    end for
18: end for
19: }
```

20: **procedure** ScanReverse($\mathcal{G}, x, y, k, V, check, pred$)
21: {
22: $ScanType = null$
23: $diag = MaxDiag$
24: **while** $diag \geq MinDiag$ **do**
25:     $lo = \min\{i : check(i, diag - i) > 0\}$
26:     $hi = \max\{i : check(i, diag - i) > 0\}$
27:     $i = lo$
28:     $j = diag - i$
29:     **while** ($i \leq hi$ **or** $check(i, j) > 0$) **do**
30:       **if** ($check(i, j) > 0$ **and** $V(i, j) + \text{LB}(|x - i| + |y - j|) < V(x, y)$) **then**
31:         **for** $\Delta i = 1, ..., \lfloor k/2 \rfloor$ **do**
32:           $v = V(i, j) + \frac{\lambda_{ij}/\mu_{ij}}{\lambda_{ij} + \mu_{ij}} + s + c(2\Delta i \cdot h, 0) + 2\Delta i \cdot t$
33:           **if** $v < V(i + \Delta i, j - \Delta i)$ **then**
34:             $V(i + \Delta i, j - \Delta i) = v$
35:             $pred(i + \Delta i, j - \Delta i) = (i, j)$
36:             $check(i + \Delta i, j - \Delta i) = 2$
37:             $ScanType = forward$
38:           **end if**
39:         **end for**
40:       **end if**
41:       $i = i + 1$
42:       $j = j - 1$
43:     **end while**
44:     $i = hi$
45:     $j = diag - i$
46:     **while** ($i \geq lo$ **or** $check(i, j) > 0$) **do**
47:       **if** ($check(i, j) > 0$ **and** $V(i, j) + \text{LB}(|x - i| + |y - j|) < V(x, y)$) **then**
48:         $check(i, j) = check(i, j) - 1$
49:         **for** $\Delta i = 1, ..., \lfloor k/2 \rfloor$ **do**
50:           $v = V(i, j) + \frac{\lambda_{ij}/\mu_{ij}}{\lambda_{ij} + \mu_{ij}} + s + c(2\Delta i \cdot h, 0) + 2\Delta i \cdot t$
51:           **if** $v < V(i - \Delta i, j + \Delta i)$ **then**
52:             $V(i - \Delta i, j + \Delta i) = v$
53:             $pred(i - \Delta i, j + \Delta i) = (i, j)$
54:             $check(i - \Delta i, j + \Delta i) = 2$
55:             $ScanType = forward$
56:           **end if**
57:         **end for**
58:         **for all** $(\Delta i, \Delta j) : \Delta i + \Delta j < 0, \ |\Delta i| + |\Delta j| \leq k$ **do**
59:           $v = V(i, j) + \frac{\lambda_{ij}/\mu_{ij}}{\lambda_{ij} + \mu_{ij}} + s + c((|\Delta i| + |\Delta j|) \cdot h, 0) + (|\Delta i| + |\Delta j|) \cdot t$
60:           **if** $v < V(i + \Delta i, j + \Delta j)$ **then**
61:             $V(i + \Delta i, j + \Delta j) = v$
62:             $pred(i + \Delta i, j + \Delta j) = (i, j)$
63:             $check(i + \Delta i, j + \Delta j) = 2$
64:             $ScanType = forward$
65:             $MinDiag = \min\{MinDiag, diag + \Delta i + \Delta j\}$
66:           **end if**
67:         **end for**
68:       **end if**
69:     **end while**
70:     $diag = diag - 1$
71: **end while**
72: }

```
73: procedure ScanForward($\mathcal{G}, x, y, k, V, check, pred$)
74: {
75:   $ScanType = null$
76:   $diag = MinDiag$
77:   while $diag \leq MaxDiag$ do
78:      $lo = \min\{i : check(i, diag - i) > 0\}$
79:      $hi = \max\{i : check(i, diag - i) > 0\}$
80:      $i = lo$
81:      $j = diag - i$
82:      while ($i \leq hi$ or $check(i, j) > 0$) do
83:         if ($check(i, j) > 0$ and $V(i, j) + \mathrm{LB}(|x - i| + |y - j|) < V(x, y)$) then
84:            for $\Delta i = 1, ..., \lfloor k/2 \rfloor$ do
85:               $v = V(i, j) + \frac{\lambda_{ij}/\mu_{ij}}{\lambda_{ij} + \mu_{ij}} + s + c(2\Delta i \cdot h, 0) + 2\Delta i \cdot t$
86:               if $v < V(i + \Delta i, j - \Delta i)$ then
87:                  $V(i + \Delta i, j - \Delta i) = v$
88:                  $pred(i + \Delta i, j - \Delta i) = (i, j)$
89:                  $check(i + \Delta i, j - \Delta i) = 2$
90:                  $ScanType = reverse$
91:               end if
92:            end for
93:         end if
94:         $i = i + 1$
95:         $j = j - 1$
96:      end while
97:      $i = hi$
98:      $j = diag - i$
99:      while ($i \geq lo$ or $check(i, j) > 0$) do
100:        if ($check(i, j) > 0$ and $V(i, j) + \mathrm{LB}(|x - i| + |y - j|) < V(x, y)$) then
101:           $check(i, j) = check(i, j) - 1$
102:           for $\Delta i = 1, ..., \lfloor k/2 \rfloor$ do
103:              $v = V(i, j) + \frac{\lambda_{ij}/\mu_{ij}}{\lambda_{ij} + \mu_{ij}} + s + c(2\Delta i \cdot h, 0) + 2\Delta i \cdot t$
104:              if $v < V(i - \Delta i, j + \Delta i)$ then
105:                 $V(i - \Delta i, j + \Delta i) = v$
106:                 $pred(i - \Delta i, j + \Delta i) = (i, j)$
107:                 $check(i - \Delta i, j + \Delta i) = 2$
108:                 $ScanType = reverse$
109:              end if
110:           end for
111:           for all $(\Delta i, \Delta j) : \Delta i + \Delta j > 0, \ |\Delta i| + |\Delta j| \leq k$ do
112:              $v = V(i, j) + \frac{\lambda_{ij}/\mu_{ij}}{\lambda_{ij} + \mu_{ij}} + s + c((|\Delta i| + |\Delta j|) \cdot h, 0) + (|\Delta i| + |\Delta j|) \cdot t$
113:              if $v < V(i + \Delta i, j + \Delta j)$ then
114:                 $V(i + \Delta i, j + \Delta j) = v$
115:                 $pred(i + \Delta i, j + \Delta j) = (i, j)$
116:                 $check(i + \Delta i, j + \Delta j) = 2$
117:                 $ScanType = reverse$
118:                 $MaxDiag = \max\{MaxDiag, diag + \Delta i + \Delta j\}$
119:              end if
120:           end for
121:        end if
122:     end while
123:     $diag = diag + 1$
124:  end while
125: }
```

33

References

[1] Artmeier, A., Haselmayr, J., Leucker, M., & Sachenbacher, M. (2010). The shortest path problem revisited: Optimal routing for electric vehicles. *Lecture Notes in Computer Science*, 6359, 309-316.

[2] Eisner, J., Funke, S., & Storandt, S. (2011). Optimal route planning for electric vehicles in large networks. *Proceedings of the 25th Association for the Advancement of Artificial Intelligence Conference.*

[3] Electric vehicle charging station locations. (2014). U.S. Department of Energy, Office of Efficiency and Renewable Energy, Alternative Fuels Data Center. Retrieved from `http://www.afdc.energy.gov/fuels/electricity_locations.html`

[4] Ferrucci, F. (2013). *Pro-active Dynamic Vehicle Routing: Real-time Control and Request-forecasting Approaches to Improve Customer Service.* Physica-Verlag.

[5] Fuel Economy Guide. (2014). U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, and U.S. Environmental Protection Agency. DOE/EE-0957. Retrieved from `http://www.fueleconomy.gov/feg/pdfs/guides/FEG2014.pdf`

[6] Khuller, S., Malekian, A., & Mestre, J. (2007). To fill or not to fill: The gas station problem. *Proceedings of the 15th Annual European Symposium on Algorithms*, 4698, 534-545.

[7] Klabjan, D. & Sweda, T. (2011). The nascent industry of electric vehicles. *Wiley Encyclopedia of Operations Research and Management Science.*

[8] Klampfl, E., Gusikhin, O., Theisen, K., Liu, Y., & Giuli, T. J. (2008). Intelligent refueling advisory system. White paper.

[9] Laur, D. (2013, April 10). Fast charging to 100%. *My Nissan Leaf Forum.* Retrieved from `http://www.mynissanleaf.com/viewtopic.php?f=27&t=12263&start=10#p282113`

[10] Lin, S. H., Gertsch, N., & Russell, J. R. (2007). A linear-time algorithm for finding optimal vehicle refueling policies. *Operations Research Letters*, 35, 290-296.

[11] Lin, S. H. (2008). Finding optimal refueling policies: A dynamic programming approach. *Journal of Computing Sciences in Colleges*, 23, 272-279.

[12] Lin, S. H. (2008). Finding optimal refueling policies in transportation networks. *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, 5034, 280-291.

[13] Millner, A. (2010). Modeling lithium ion battery degradation in electric vehicles. *Proceedings of the Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*, 349-356.

[14] Nissan Leaf Digital Brochure. (2014). Nissan USA. Retrieved from `http://www.nissanusa.com/content/dam/nissan/ request-brochure/en/2014/pdf/2014-nissan-leaf-en.pdf`

[15] Sachenbacher, M., Leucker, M., Artmeier, A., & Haselmayr, J. (2011). Efficient energy-optimal routing for electric vehicles. *Proceedings of the 25th Association for the Advancement of Artificial Intelligence Conference.*

[16] Serrao, L., Onori, S., Sciarretta, A., Guezennec, Y., & Rizzoni, G. (2011). Optimal energy management of hybrid electric vehicles including battery aging. *Proceedings of American Control Conference*, 2125-2130.

[17] Suzuki, Y. (2008). A generic model of motor-carrier fuel optimization. *Naval Research Logistics*, 55, 737-746.

[18] Suzuki, Y. (2009). A decision support system of dynamic vehicle refueling. *Decision Support Systems*, 46, 522-531.

[19] Suzuki, Y., & Dai, J. (2013). Decision support system of truck routing and refueling: A dual-objective approach. *Decision Sciences*, 44, 817-842.

[20] Sweda, T. M. & Klabjan, D. (2012). Finding minimum-cost paths for electric vehicles. *Proceedings of the 1st IEEE International Electric Vehicle Conference*, 1-4.

[21] Sweda, T. M., Dolinskaya, I. S., & Klabjan, D. (2014). Optimal recharging policies for electric vehicles. Working Paper 14-01, Department of Industrial Engineering and Management Sciences, Northwestern University.

[22] Thomas, B. W., & White III, C. C. (2004). Anticipatory route selection. *Transportation Science*, 38, 473-487.

[23] Thomas, B. W., & White III, C. C. (2007). The dynamic shortest path problem with anticipation. *European Journal of Operations Research*, 176, 836-854.