

# Intention-Aware Routing to Minimise Delays at Electric Vehicle Charging Stations\*

**Mathijs M. de Weerd**  
Delft University of Technology  
m.m.deweerd@tudelft.nl

**Enrico H. Gerding, Sebastian Stein,  
Valentin Robu, and Nicholas R. Jennings**  
University of Southampton  
{eg,ss2,vr2,nrj}@ecs.soton.ac.uk

## Abstract

En-route charging stations allow electric vehicles to greatly extend their range. However, as a full charge takes a considerable amount of time, there may be significant waiting times at peak hours. To address this problem, we propose a novel navigation system, which communicates its intentions (i.e., routing policies) to other drivers. Using these intentions, our system accurately predicts congestion at charging stations and suggests the most efficient route to its user. We achieve this by extending existing time-dependent stochastic routing algorithms to include the battery's state of charge and charging stations. Furthermore, we describe a novel technique for combining historical information with agent intentions to predict the queues at charging stations. Through simulations we show that our system leads to a significant increase in utility compared to existing approaches that do not explicitly model waiting times or use intentions, in some cases reducing waiting times by over 80% and achieving near-optimal overall journey times.

## 1 Introduction

The expected increase in the number of electric vehicles (EVs) necessitates novel solutions for managing the infrastructure required to charge these vehicles [R.A.E., 2010]. One of the most significant potential bottlenecks is the congestion at charging stations. This is likely to occur because fully charging an EV currently takes between half an hour and several hours and, moreover, EVs need to be charged frequently to complete longer trips. In order to address this challenge, new navigation systems need to be designed that predict congestion at charging stations based on stochastic information about arrivals, and then suggest to EV owners the most efficient route and station, in order to minimise both travel time and expected delays due to charging.

Within the transportation literature, the problem of optimal routing under uncertainty has been studied for several years. In particular, Hall [1986] and later Gao and Chabini [2006]

show that optimal policies can be computed even for settings where the stochastic information is not only link-dependent, but also time-dependent (i.e., travel times are conditional on the time of day). These approaches compute *routing policies*, i.e., time and location-dependent directions, to deal with such situations where travel times are not known in advance. However, they currently do not consider the specific requirements of EVs, which have a limited charge and may need to route past charging stations to reach their destination.

The problem of routing agents through a network is also studied using game-theoretic approaches. In this vein, Gawron [1998] studies a model in which each driver's choice over alternative routes is given by a discrete probability distribution and finds a dynamic equilibrium through simulation. However, these approaches take a centralised perspective and, moreover, abstract away from the details of the problem we consider related to EVs, such as their charging state and delays at charging stations.

A crucial problem, from an EV perspective, is *coordinating* en-route charging. Since charging EVs takes time and the capacity at charging stations is limited, waiting times can contribute significantly to the total journey time. For example, if one additional vehicle arrives at a station with a single charging point, this could immediately result in a 30-minute delay to all subsequent arrivals. Indeed, this problem has been recognised before. In particular, Qin and Zhang [2011] propose an approach where congested charging stations divert EVs by placing reservations at less congested stations on the same route. Kim *et al.* [2012] develop a constraint-based scheduler for charging stations to prevent peak consumption. These existing solutions are based on reserving a specific time slot. However, travel times are typically uncertain, and the distribution of such travel times changes throughout the day. Because of this, delays give rise to changes in the arrival time, but could also necessitate re-routing to a different charging station, invalidating any reservations.

To address this problem, we propose a novel Intention-Aware Routing System (IARS), which is implemented as a software agent embedded in the navigation system of an EV. Crucially, this agent autonomously exchanges *intentions* (i.e., routing decisions) with other IARS agents. These intentions constitute probabilistic information about *which* stations the EVs will visit and *when*, thereby allowing the agent to accurately predict congestion (and thus waiting times) at charg-

\*This work was supported by the ORCHID (orchid.ac.uk) and iDEaS projects (ideasproject.info).

ing stations and compute the most efficient route for the EV owner. Our proposed system employs and extends a range of techniques from the field of artificial intelligence to deal with the highly dynamic and uncertain environment it operates in. In more detail, it makes the following contributions to the current state of the art:

- We formalise the optimal EV routing problem with en-route charging stations. This is a challenging problem, because both travel and waiting times are stochastic and time-dependent. Furthermore, the problem is highly constrained by the current state of charge of an EV and the need to route via charging stations.

- We propose the first EV routing system that uses intentions to produce accurate predictions of waiting times at charging stations. A key contribution of this system is its novel approach to fusing a range of information sources that vary in their precision — from information about historical and recent EV arrivals at stations (including from those EVs that do not communicate their intentions), to the probabilistic information encoded in the agents’ intentions.

- We show experimentally that IARS leads to a significantly higher utility than state-of-the-art routing algorithms that rely only on historical information about road travel times (as used by some modern navigation devices). In some cases, our approach leads to an over 80% improvement in waiting times and a more than 50% reduction in overall journey times. As IARS also reasons about agents that do not use intentions, we demonstrate that even when only a small proportion of EV drivers use IARS, they achieve a significantly higher utility than those that do not.

The remainder of the paper is structured as follows. In Section 2 we formalise the routing problem with en-route charging. Then, in Section 3, we explain IARS and in Section 4 describe our experimental results. Finally, Section 5 concludes.

## 2 Model

First we introduce our model of stochastic time-dependent routing for EVs, where roads and charging stations are abstractly represented by probability distributions of their waiting time. We then go on to model the charging stations and the waiting times explicitly, which enables us to derive the waiting time probabilities when computing the policies, taking into account the intentions of other EVs.

### 2.1 The EV Routing Problem

We model an EV routing domain by  $\langle V, E, T, P, S, C \rangle$ , with directed edges  $e = \langle v_i, v_j \rangle \in E$  and vertices  $v_i, v_j \in V$ . Edges represent either roads or charging stations, denoted by  $E_{\text{stations}} \subset E$  and  $E_{\text{roads}} \subset E$  respectively. Roads and charging stations incur a probabilistic amount of travel or waiting time, described by a probability mass function  $P$  (more details below). These travel and waiting times vary depending on the time of the day, and  $T = \{1, 2, \dots, t_{\text{max}}\}$  denotes a finite set of time points (e.g., within a day, or over a week). Roads furthermore incur a cost to EVs in terms of power usage, whereas charging stations reset the EV battery to its maximum capacity level (we assume that a battery is always fully charged at a station). To this end, we introduce a finite

set of possible charging *states*  $S = \{0, 1, \dots, s_{\text{max}}\}$ , where a state represents the current capacity of the battery, and  $s_{\text{max}}$  denotes a fully charged battery. Furthermore, we introduce function  $C$ , where  $C(e) \in S$  is the (deterministic) charging cost for edge  $e \in E_{\text{roads}}$ .<sup>1</sup> This charging cost is deducted from the current state of charge when the edge is traversed.

We consider time-dependent stochastic travel and waiting times and treat them as stochastically independent. That is, conditional on the time of day, the distributions at edges are uncorrelated, and we do not take into account that these distributions may be updated over time. This is common in the stochastic routing literature [Hall, 1986]. Formally,  $P(t_b - t_a | e, t_a)$  denotes the probability mass function of the travel/waiting time at edge  $e = \langle v_a, v_b \rangle \in E$ , where  $t_a \in T$  denotes the arrival time at vertex  $v_a$ , and  $t_b \in T, t_b \geq t_a$  the arrival time at vertex  $v_b$ . Thus, when  $e$  is a road, then  $t_b - t_a$  is the travel time, and when  $e$  is a charging station,  $t_b - t_a$  is the combined waiting and charging time.

Given this, the problem is to find an optimal *routing policy*  $\pi^*$  which maximises the agent’s expected utility without running out of charge at any point during the journey. Formally, a routing policy is a function  $\pi : V \times T \times S \rightarrow V$  which gives, for each vertex, each possible realisation of the arrival time, and each state of charge, the next vertex. Then, given a policy  $\pi$ , our current position  $v_c \in V$ , current time  $t_c \in T$ , current state of charge  $s_c \in S$  and final destination  $v_{\text{dest}} \in V$ , the first edge to follow is equal to  $e_c = (v_c, \pi(v_c, t_c, s_c))$  and the expected utility for the policy  $\pi$  can be computed using the following recursive formulation:

$$\begin{aligned} \text{EU}(e_c = (v_c, w), t_c, s_c | \pi) = & \\ & \begin{cases} -\infty & \text{if } s_c \leq 0 \\ \sum_{\Delta t \in T} P(\Delta t | e_c, t_c) \cdot U(t_c + \Delta t, s') & \text{if } w = v_{\text{dest}} \\ \sum_{\Delta t \in T} P(\Delta t | e_c, t_c) \cdot \\ \text{EU}((w, \pi(w, t_c + \Delta t, s')), t_c + \Delta t, s' | \pi) & \text{otherwise} \end{cases} \end{aligned}$$

where  $s' = \text{SOC}(e_c, s_c)$  determines the new state of charge when traversing edge  $e_c$ , i.e.,  $\text{SOC}(e_c, s_c) = s_c - C(e_c)$  if  $e_c \in E_{\text{roads}}$  and  $\text{SOC}(e_c, s_c) = s_{\text{max}}$  if  $e_c \in E_{\text{stations}}$ . Furthermore,  $U(t_c, s_c)$  is the agent’s utility function for a given arrival time  $t_c$  and a state of charge  $s_c$  on arrival such that  $U(t_c, s_c) = -\infty$  if  $s_c < 0$ . Otherwise, for example,  $U(t_c, s_c) = -t_c$ ; then maximising the expected utility means minimising the expected time of arrival.

The *optimal* policy is then given by  $\pi^*(v_c, t_c, s_c) = \arg \max_{\pi \in \Pi} \text{EU}((w, \pi(v_c, t_c, s_c)), t_c, s_c | \pi)$ , where  $\Pi$  is the set of all valid policies.

### 2.2 Charging Stations Model

So far, we have modelled the waiting times at both edges and stations using abstract probability distributions, and we have not discussed where these distributions actually come from. Now, a main contribution of our paper is to combine historic information and intentions (policies) of other agents to derive the waiting time distributions at charging stations (we assume

<sup>1</sup>Since we compute the route for each vehicle separately, it is straightforward to extend this model by allowing a different function  $C$  and a different  $s_{\text{max}}$  for each type of EV.

- 1 **user input:** destination  $v_{i,dest}$
- 2 initialise  $v_i, s_i,$  and  $t_i$  to the current state
- 3 compute  $P^{arr}$  based on historical arrivals
- 4 **while**  $v_i \neq v_{i,dest}$  **do**
- 5   **input:** states  $v_j, s_j, t_j,$  intentions  $\pi_j, \forall j \in I \setminus \{i\},$   
and new arrivals
- 6   compute  $P_j^{arr}(e, t), \forall j \in I$
- 7   compute  $P(\Delta t|e, t), \forall e \in E_{stations}$
- 8   compute policy  $\pi_i$
- 9   **output:** intention  $\pi_i$  and current state  $v_i, s_i, t_i$
- 10 execute policy and update  $v_i, s_i, t_i$

**Algorithm 1:** Intention-aware routing for an EV

that the distributions for roads are derived using existing approaches, such as [Eglese *et al.*, 2006], since they are influenced less by individual EVs). However, to do so, we first need to explicitly model the queues at the charging stations. In Section 3, we then show how we use this model to derive the waiting time probability distributions using a principled approach that includes the intentions of other agents.

The queueing model is as follows. We assume that each station  $e \in E_{stations}$  has a fixed capacity,  $cap_e$ , due to space or electricity network constraints. This capacity is the maximum number of vehicles that can charge simultaneously. Furthermore, for simplicity, we assume that the time to (fully) charge a vehicle, denoted by  $t^{charge} \in T$ , is fixed (although it is straightforward to extend the model to stochastic or charge-dependent times). We assume a first-come-first-served queueing model when the station is at full capacity and that there is no queue before time  $t = 1$ . Finally, if several EVs arrive at the same time, we assume they arrive in the order of their ID.

### 3 Intention-Aware Routing System

The IARS optimises the policy of an EV using the intentions of other EVs. The main feature of this system is that it is able to route using only historical information when no other information is available, but then gradually replaces this information with more accurate information about vehicles whose intentions become known or change, and as vehicles actually arrive at the stations. Algorithm 1 presents an overview of the steps involved. In detail, the journey starts by the user, denoted by  $i$ , entering his or her destination,  $v_{i,dest}$ . The system is then initialised based on the current location, state of charge and departure time (line 2). Before it can compute the optimal policy (line 8), it needs to process the historical information about arrivals of vehicles at stations, and combine these with the currently known intentions of other users in the system and actual arrivals at the stations (lines 3, 6, and 7, detailed in the subsections below), where  $I$  denotes the set of EVs whose intentions are currently known. Importantly, since the system only knows intentions of users who have already entered their destination, the set  $I$  is not fixed and can change over time with users entering and leaving the system dynamically. For this reason, and because the available information is continually updated (line 5), the probability distributions and optimal policies are also updated at each new vertex (i.e., the decision points). We now describe the computation steps, lines 3, 6, 7, and 8 in detail.

### 3.1 Historical Arrival Probabilities

A possible approach to compute waiting times probabilities is to keep track of the historic waiting times, and simply take the average waiting time for each station and time point. However, this approach does not readily allow information of individual vehicles to be integrated as and when this becomes available. We therefore take a different approach which keeps track of *when* and *where* (i.e., at what station) vehicles arrive for charging.<sup>2</sup> We then aggregate these historical arrivals to compute the probabilities  $P^{arr}(e, t)$  which gives, for an average EV, the probability that he or she arrives at station  $e \in E_{stations}$  at time  $t \in T$ . In Section 3.3 we then show how this distribution is used to estimate the queues at the stations, and compute the waiting times.

### 3.2 Intention-Based Arrival Probabilities

The historical arrival probabilities represent the likelihood of a vehicle arriving at a certain station at a certain time, without having any additional information about these vehicles. However, a key aspect of our approach is that, for those vehicles with intentions, we can derive much more accurate estimates about which station they are going to visit, based on their current state and their policy (note that the information about station visits is still probabilistic since the actual route and arrival time depends on the realised travel times). To this end, we show how to derive  $P_i^{arr}(e, t)$ , the probability that EV  $i$  is at station  $e \in E_{stations}$  at time  $t \in T$ .

In detail, given  $\pi_i$ , the current state  $(v_i, t_i, s_i)$  of  $i$ , and known travel time distributions at road edges  $P(\Delta t|(v, w), t)$ , we can compute  $P_i^{arr}$  via summing up over  $s \in S$  of  $P_i^{arr}(v, t, s)$ , the probabilities that  $i$  arrives at a particular vertex  $v$  at a particular time  $t$  with charging state  $s$ , using the dynamic program below:

$$\begin{aligned}
 &P_i^{arr}(v, t, s) \leftarrow 0 \text{ for all } v, t, s, \text{ but } P_i^{arr}(v_i, t_i, s_i) \leftarrow 1 \\
 &P_i^{arr}(e, t) \leftarrow 0 \text{ for all } t \in T, e \in E_{stations} \\
 &\text{for } t \leftarrow 1, 2, \dots, t_{max} \text{ do} \\
 &  \text{forall the } v \in V \text{ and } s \in S \text{ do} \\
 &    \text{if } P_i^{arr}(v, t, s) > 0 \text{ then} \\
 &       $(v, w) \leftarrow \pi(v, t, s)$  \\
 &      \text{if } (v, w) \in E_{stations} \text{ then} \\
 &        \text{add } P_i^{arr}(v, t, s) \text{ to } P_i^{arr}((v, w), t) \\
 &      \text{for } \Delta t \in T \text{ do} \\
 &        \text{add } P(\Delta t|(v, w), t) \text{ to} \\
 &           $P_i^{arr}(w, \Delta t + t, \text{SOC}((v, w), s))$ 
 \end{aligned}$$

### 3.3 Computing Waiting Times Probabilities

We now discuss the main part of the system and show how to compute the waiting times probability mass function,  $P(\Delta t|e, t)$ , by combining the historical information,  $P^{arr}$ , with known arrivals so far, and with the intentions-derived probabilities,  $P_i^{arr}$ ,  $i \in I$ , where  $I$  is the set of EVs who have (so far) reported their intentions to the system. We let  $n$  denote the total number of unique vehicles that have charged in the past (across stations), including both ones that use the system, and ones that do not. For simplicity, we assume that each EV charges en-route at most once a day, although having

<sup>2</sup>In practice, this can be achieved, for example, through sensors and/or credit card payment information at stations.

a single vehicle charge multiple times can be approximated by considering these are different vehicles (in which case  $n$  is the maximum number of charges per day). Furthermore, let  $m$  denote the number of vehicles who have already charged today, and  $I' \subseteq I$  those vehicles with known intentions who still need to charge (i.e., they are visiting a station with non-zero probability). Given this, there are  $n - m$  EVs that may still choose to charge, of which  $|I'|$  we know their intentions.

We then approximate the probability mass function using *Monte-Carlo simulation* as follows. For each sample, we independently draw for  $n - m - |I'|$  vehicles a pair  $\langle e, t \rangle \in (E_{\text{stations}} \times T) \cup \{\perp\}$  according to the probabilities  $P^{arr}(e, t | \text{not charged})$ , i.e., the arrival conditional on not having charged before (where the probability of charging before the current time is zero). Here,  $P^{arr}(\perp) = 1 - \sum_{e \in E_{\text{stations}}, t \in T} P^{arr}(e, t | \text{not charged})$  is the probability that the EV does not charge at all. Similarly, we draw a single pair  $\langle e, t \rangle$  from each distribution  $P_i^{arr}$  (note that we do not need to compute the conditional distribution, since it has already been updated). Finally, we add the EVs that have already arrived today with probability 1. Then, starting from  $t = 1$ , we simulate the queues at each station based on the model described in Section 2.2 until the end of the day, and measure the waiting times for each future time point. This process is repeated for a number of times (in the experiments in Section 4 we use 100 samples), and  $P(\Delta t | e, t)$  is estimated by averaging the waiting times at each station and time slot.

### 3.4 Computing the Optimal Policy

The final part is the computation of the routing policy. Since for every computation of EU the policy  $\pi_i$  is required only for times strictly later than  $t_i$  (we assume  $\Delta t > 0$ ), the optimal policy can be computed using dynamic programming in line with work on Markov decision processes [Puterman, 1994] based on the following recursive definition: if  $v_i = v_{dest}$  or  $s_i \leq 0$ , then there is no good decision, and otherwise  $\pi^*(v_i, t_i, s_i) = \arg \max_{\{e | (v_i, w) \in E\}} \text{EU}(e, t_i, s_i | \pi^*)$ . All computations described above can be done in  $\mathcal{O}(|T|^2 \cdot |V| \cdot |S| \cdot |E|)$ .

## 4 Experiments

In this section, we experimentally evaluate our intention-aware routing system in a wide range of settings. The purpose of this is to establish and quantify the potential benefits of 1) modelling station waiting times and 2) incorporating other agents' intentions into routing decisions. For ease of presentation, we assume that all agents wish to minimise their arrival time at the destination, and therefore our primary measure of performance is the average journey time of individual agents. In the following, we first describe the benchmarks we test, provide details of our setup and then present the results.

### 4.1 Benchmarks Solutions

We evaluate a range of navigation systems in this section:

- **IARS**: Our proposed intention-aware routing system, which is the main contribution of this paper.
- **MIN**: A system that always minimises the expected journey time. As such, it simulates existing state-of-the-art navigation systems.

- **LOGIT( $\lambda$ )**: This is a randomised variant of MIN. We include this, because agents employing MIN on similar source and destination pairs will often follow the same routes, exacerbating congestion at charging stations. While this is an inherent problem with current routing systems, we are interested in whether occasional randomisation may alleviate this. To achieve this, we use an approach where the probability of selecting an alternative is directly related to the expected utility of that same alternative. This is in line with work on the logit *agent* quantal response equilibrium [McKelvey and Palfrey, 1998] and is defined as follows. Given a  $\lambda \in [0, \infty]$  the probability of selecting an edge  $e$  is defined as:

$$P(e | v_c, t_c, s_c) = \frac{e^{\lambda \cdot \text{EU}(e, t_c, s_c | \pi')}}{\sum_{\{e' | (v_c, w) \in E\}} e^{\lambda \cdot \text{EU}(e', t_c, s_c | \pi')}} \quad (1)$$

The policy  $\pi'(v_c, t_c, s_c)$  is then just drawn from this distribution, and the expected utilities are computed knowing that this distribution is used in future time steps, i.e.,  $\text{EU}((u, v_c), t_c, s_c | \pi') = \sum_{\{e' | (v_c, w) \in E\}} P(e' | v_c, t_c, s_c) \cdot \text{EU}(e', t_c, s_c | \pi')$ .

We also attach a parameter FALSE or TRUE to each approach other than IARS, which determines whether the system models waiting times at charging stations using *historical arrivals* (TRUE), or always assumes the minimum charging time (FALSE). As such, MIN(FALSE), for example, always picks the shortest route and does not model charging delays (which may occur when using current standard GPS routing systems in an EV), while LOGIT(100, TRUE) uses historical arrivals to estimate queueing times, but sometimes randomly deviates from the optimal path.

### 4.2 Experimental Setup

Throughout our experiments, we assume that charging takes 30 minutes, which is realistic for current state-of-the-art charging stations (modelling 10 minutes as one time step). We also run each setting for 20 simulated days, in order to allow agents to learn and respond to the behaviour of others.<sup>3</sup> Whenever agents use Monte Carlo simulation to estimate arrival probabilities, we take 100 samples, which leads to good results in practice. We also repeat all experiments 1,000 times and report 95% confidence intervals.

We examine two separate scenarios in this section, in order to verify that similar trends hold in different graph topologies and problem sizes. Both scenarios are shown in Figure 1. Here, EVs leave from vertices labelled 'S' (source) and wish to travel to vertices labelled 'D' (destination), which are chosen uniformly at random between the alternatives in the second scenario. Charging stations are indicated by looping edges, with their charging capacities given by the label on the adjacent vertex. We set charging costs appropriately to ensure that all EVs must charge at least once, and, as we are interested in how the systems cope with congestion at peak times, we assume all vehicles leave at time  $t = 1$  (we obtained similar results when vehicles leave gradually).

More specifically, the first of these scenarios, the *bottleneck* setting, focuses only on the decision of which charging

<sup>3</sup>This is a conservative choice — in practice, the results typically do not change significantly after the first five days.

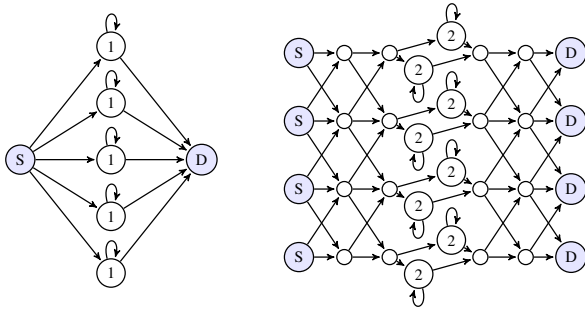


Figure 1: Routing graphs *bottleneck* (left) and *grid* (right).

stations to route to and assumes that all agents share the same source and destination. Here, non-station edges have probabilistic travel times, which we generate by drawing five samples from the set  $\{1, 2, \dots, 12\}$  (with replacement). Then we attach a value drawn uniformly at random from  $[0, 1]$  to each unique sample and normalise these to sum to 1, in order to obtain a probability mass function.<sup>4</sup> This is a realistic setting for the case where several potential routes exist to a popular destination, e.g., different roads between two large cities or from a commuter town to the commercial centre of a city. It represents an extreme setting, as agents need to commit to a station on departure, some of which will clearly be more desirable than others in the absence of congestion and may therefore become bottlenecks. Here, we simulate 20 EVs.

The second scenario, the *grid* setting, represents a case where agents have diverse sources and destinations and can potentially change their policy based on new information before reaching a station (if using IARS). Thus, it models a realistic road network, e.g., between several cities or other points of interest. Here, the travel time distribution on each road edge is generated by drawing two samples from  $\{1, 2, \dots, 4\}$ . We simulate 50 EVs, representing a larger, more complex setting.<sup>5</sup> Note the average travel time from a source to a random charging station (and again to the destination) is between 65 minutes for *bottleneck*, and 75 minutes for *grid*. This is reasonable, given the current range of EVs.

### 4.3 Results for Overall Performance

In our first set of experiments, we compare the performance of all systems in the two settings, as shown in Figure 2. As we show in Section 4.4,  $\lambda = 10$  and  $\lambda = 100$  perform well in these settings, respectively, and so we only show their performance here. For the *bottleneck* setting we also plot a centrally computed optimal solution. As a globally optimal algorithm is computationally not feasible in the *grid* setting (due to the large number of vehicles and possible paths), we give a lower bound on the journey time here, which is based on a simpler

<sup>4</sup>For ease of presentation and because we focus on charging stations in this paper, these are not time-dependent.

<sup>5</sup>We stress that significantly larger settings are similarly feasible. To illustrate this, simulating 1,000 EVs with IARS arriving over the course of a day in the *grid* setting takes just over two minutes. We choose 50 EVs here for practical reasons, to allow us to explore a large parameter space and collect statistically significant data. The general trends are the same as in larger settings.

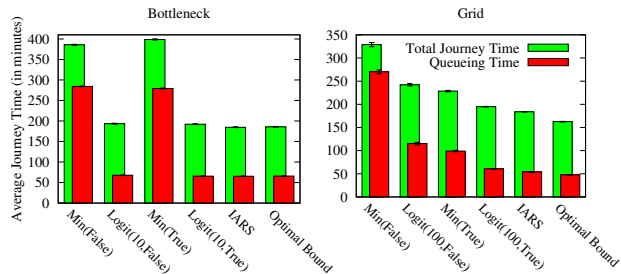


Figure 2: Average journey and waiting times for different systems in the *bottleneck* (left) and the *grid* (right) settings.

graph that combines all charging stations into a single one with an appropriately higher capacity. This is a very optimistic lower bound, because it allows a simple optimal strategy, where vehicles ignore congestion and choose the path with the shortest expected travel time.

Several interesting trends emerge here. First, in the *bottleneck* setting, both MIN approaches perform badly, leading to an average journey time of around 386 and 399 minutes (for MIN(FALSE) and MIN(TRUE), respectively). This is more than twice the optimal with 186 minutes. The reason for this is that the MIN approaches always choose the one route that minimises their travel time (in expectation), but as all agents act on the same information, they pick the same path, leading to a single extremely congested station. This is evidenced by the high proportion of time spent queueing rather than travelling. Surprisingly, performance decreases even further when an explicit model of waiting times is used by the MIN approach. This is because the agents learn that the first station was highly congested, but then move in tandem to the next best option, resulting in the same queues, but longer travel times. Clearly, this highlights the perils of using a simple travel time minimisation approach in settings where agents have similar requirements.

The remaining approaches in the *bottleneck* setting all perform surprisingly well. Our proposed IARS system achieves the same performance as the globally optimal benchmark, while even the simpler randomised LOGIT approaches lead to an average journey time of 192 minutes.

The trends in the *grid* setting are slightly different. Here, MIN(FALSE) without queueing model still achieves an overall bad performance with an average journey time of 329 minutes (a 101% increase over the optimal bound of 163 minutes), most of which is spent queueing again. However, this is significantly improved by including an explicit model of station waiting times using historical arrivals (223 minutes, 38% increase), because it allows the agent to reason about queues (which are more heterogeneous in this setting due to the variable sources and destinations). Also, adding randomisation (242 minutes, 48% increase over the bound) is beneficial in this setting, because it avoids congestion at otherwise more desirable stations. Combining both, the LOGIT(100,TRUE) achieves an average travel time of 195 minutes (20% increase). Finally, the IARS achieves an even higher performance, with an average journey time of 184 minutes (13% more than the optimal bound).

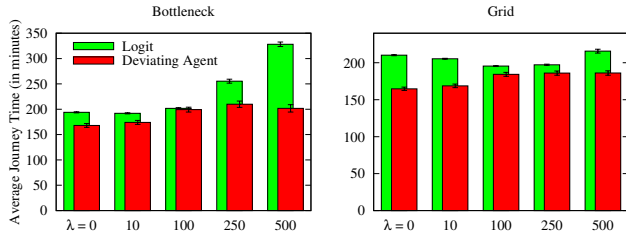


Figure 3: Average journey times for varying  $\lambda$  parameters in the *bottleneck* setting (left) and the *grid* setting (right).

#### 4.4 Results for LOGIT( $\lambda$ )

So far, LOGIT( $\lambda$ ) appears to be a promising alternative to IARS. To examine this in more detail, we show the performance of LOGIT( $\lambda$ ,TRUE) for a range of  $\lambda$  parameters in Figure 3. These results show that LOGIT is somewhat sensitive to the choice of  $\lambda$ , which raises the question of how  $\lambda$  should be chosen in real-life settings.

More significantly, however, the figure also shows the average journey time a single deviating agent would achieve by switching to the MIN(TRUE) approach (assuming all other agents use LOGIT). This is never higher than LOGIT and, in fact, typically lower. In other words, while randomisation benefits the overall system, as it disperses agents across the stations, a single agent always has an incentive to deviate and head for the station with the lowest expected journey time. Thus, LOGIT is not a viable alternative in realistic systems.

#### 4.5 Results for IARS

To further explore our proposed IARS system, we now consider settings where only a proportion of agents use IARS, while the others use MIN(TRUE) or LOGIT(100,TRUE). This is an interesting setting, because it shows how the system performance changes as intentions are gradually introduced into a system (and indeed whether it is beneficial if only a few agents use IARS). Similar to the previous section, it also investigates whether agents have an incentive to switch to (or from) IARS. In this setting, we measure the average journey time for each type of agent, but we also measure the system-wide average across the population (i.e., the *social welfare*).

The results are given in Figure 4 (we show only the *grid* settings, as the results in the *bottleneck* setting are similar). Here, several clear trends emerge: as more agents use IARS, the system-wide average journey time decreases. Furthermore, agents that use IARS always have lower average journey times than those that do not, indicating that there is a strong incentive to use IARS for all system participants. Apart from this, the graphs show two further interesting trends. First, even when only a few agents use intentions, their average journey time already dramatically decreases as they can coordinate their decisions. In fact, when others use LOGIT, the IARS agents achieve their minimum journey time when only a small proportion use the system. This is similar to the results in Figure 3, where agents can exploit inefficiencies in the strategies of other agents. The second interesting trend is that agents using MIN benefit themselves when more agents switch to IARS, but this trend is not evident for agents

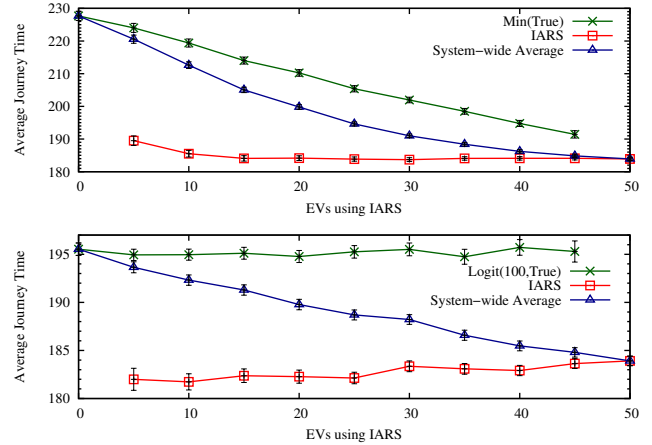


Figure 4: Average journey times as the number of EVs using IARS is varied in the *grid* setting. Top graph assumes others use MIN(TRUE), while bottom assumes LOGIT( $\lambda$ ,TRUE).

using LOGIT. This is simply because the typically high congestion at desirable stations caused by MIN is gradually reduced as fewer agents adopt this approach.

## 5 Conclusions and Future Work

This work extends both models and algorithms for stochastic time-dependent routing to take charging stations and the state of charge of EVs into account. Another main contribution is a novel approach where we combine (stochastic) intentions of other agents with historical data to obtain more accurate waiting time distributions. In order to do so, we explicitly model queues at charging stations. As part of our evaluation we defined an optimistic bound, as well as an alternative routing system based on logit that does not use intentions, but helps to prevent congestion at charging stations by randomising over options. Through extensive experiments, we established that both an explicit queueing model as well as randomisation increases social welfare, but that IARS outperforms even a combination of these systems. Moreover, agents using the randomised system have an incentive to switch to a deterministic strategy, while we showed that individual agents always achieve lower journey times with IARS.

The IARS does not require any monetary transfers. However, an interesting direction for further study is to view coordinating the en-route charging of EVs under uncertainty as a mechanism design problem. In particular, we are interested in combining our work with pricing models to be able to find more efficient solutions, such as used for charging at home [Stein *et al.*, 2012; Clement-Nyns *et al.*, 2010; Vasirani and Ossowski, 2011].

To transition our solution into a practical application, we additionally plan to extend our work with simulations with real data to obtain better insights into the quantitative advantages of deploying IARS, and it would be useful to develop heuristics for both policy computation as well as combining the distributions to improve run time and scalability.

## References

- [Clement-Nyns *et al.*, 2010] K. Clement-Nyns, E. Haesen, and J. Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on Power Systems*, 25(1):371–380, 2010.
- [Eglese *et al.*, 2006] R. Eglese, W. Maden, and A. Slater. A road timetable to aid vehicle routing and scheduling. *Computers & Operations Research*, 33(12):3508–3519, 2006.
- [Gao and Chabini, 2006] S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.
- [Gawron, 1998] C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(3):393–407, 1998.
- [Hall, 1986] R.W. Hall. The fastest path through a network with random time-dependent travel times. *Transportation science*, 20(3):182–188, 1986.
- [Kim *et al.*, 2012] H.-J. Kim, J. Lee, and G.-L. Park. Constraint-based charging scheduler design for electric vehicles. In *Intelligent Information and Database Systems*, volume 7189 of *LNCS*, pages 266–275, 2012.
- [McKelvey and Palfrey, 1998] R.D. McKelvey and T.R. Palfrey. Quantal response equilibria for extensive form games. *Experimental economics*, 1(1):9–41, 1998.
- [Puterman, 1994] M.L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994.
- [Qin and Zhang, 2011] H. Qin and W. Zhang. Charging scheduling with minimal waiting in a network of electric vehicles and charging stations. In *Proc. of the 8th ACM International Workshop on Vehicular Inter-Networking*, pages 51–60, 2011.
- [R.A.E., 2010] R.A.E. *Electric Vehicles: Charged with potential*. Royal Academy of Engineering, 2010.
- [Stein *et al.*, 2012] S. Stein, E. Gerding, V. Robu, and N.R. Jennings. A model-based online mechanism with pre-commitment and its application to electric vehicle charging. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 669–676, 2012.
- [Vasirani and Ossowski, 2011] M. Vasirani and S. Ossowski. A computational monetary market for plug-in electric vehicle charging. In *Auctions, Market Mechanisms and Their Applications*, volume 80 of *Lecture Notes of ICST*, pages 88–99, 2011.