Bid2Charge: Market User Interface Design for Electric Vehicle Charging

Sebastian Stein* ss2@ecs.soton.ac.uk
Enrico H. Gerding* eg@ecs.soton.ac.uk
Adrian Nedea* adrian.nedea@gmail.com
Avi Rosenfeld◇ rosenfa@gmail.com
Nicholas R. Jennings* nrj@ecs.soton.ac.uk

*University of Southampton, Southampton, United Kingdom
◇Jerusalem College of Technology, Jerusalem, Israel

ABSTRACT

We consider settings where owners of electric vehicles (EVs) participate in a market mechanism to charge their vehicles. Existing work on such mechanisms has typically assumed that participants are fully rational and can report their preferences accurately to the mechanism or to a software agent participating on their behalf. However, this may not be reasonable in settings with non-expert human end-users. To explore this, we compare a fully expressive interface that covers the entire space of preferences to two restricted interfaces that reduce the space of possible options. To enable this analysis, we develop a novel game that replicates key features of an abstract EV charging scenario. In two extensive evaluations with over 300 users, we show that restricting the users' preferences significantly reduces the time they spend deliberating. More surprisingly, it also leads to an increase in their utility compared to the fully expressive interface (up to 70%). Finally, we find that a reinforcement learning agent displays similar performance trends, enabling a novel methodology for evaluating market interfaces.

Keywords

Market user interface design; smart grid; electric vehicle charging

1. INTRODUCTION

Market mechanisms designed for multi-agent systems hold considerable promise for addressing emerging challenges in future electricity networks, where supply will increasingly be generated from intermittent and unreliable renewable sources, and where demand will increase due to the electrification of transportation [12]. In particular, recent years have seen a proliferation of interest in electric vehicles (EVs), generally perceived as a key technology for achieving sustainable mass transportation with low carbon emissions [14]. However, their widespread use will also place considerable strains on the existing electricity infrastructure.

To address this, previous work has proposed auction-like mechanisms for scheduling the charging of EVs [13, 4]. These achieve a high efficiency because they take the individual preferences of drivers (i.e., their availability and willingness to pay) into account when allocating a limited supply of electricity. Other work relies on real-time price signals to incentivise autonomous charging agents to shift or curtail their consumption when supply is low [11].

However, such approaches assume that the human end-users have perfect knowledge of their preferences, i.e., they can reason accurately about the value of different amounts of electricity, considering all possible, often uncertain future opportunities of using it. This is often not realistic [18, 6]. Moreover, providing the complete preferences is tedious and the associated cost could outweigh the benefits of doing so [7].

To address this challenge, there has been some work on auctions with *restricted reporting*, i.e., where bidders do not report their full preferences, but rather choose from a restricted messaging space. Such auctions can lead to equilibria with certain desirable properties, including higher revenue for the auctioneer [8, 3]. This includes settings where the messaging space is reduced to a small set of discrete options, e.g., [1] describes how to select these options to maximise either social welfare or revenue, and [2] characterises the associated loss of efficiency. Other work, e.g., [15], considers how complex preferences can be elicited through an incremental query process. However, these approaches all assume rational agents and do not evaluate the auctions with real bidders.

Another strand of work explicitly considers non-expert market participants. For example, research on hidden market design has looked at building simple interfaces that hide the rules and pricing mechanisms of a complex underlying market [16]. However, that work cannot be applied directly to EV charging, as it considers an exchange market for computational storage without financial payments. Related to this, [17] has investigated market user interface design. That work focuses on simplifying complex market interactions by asking users to select from a discrete set of options. However, it considers a setting with posted prices and is significantly simpler than the EV setting. Crucially, neither [16] nor [17] provides a direct comparison to fully expressive interfaces, which leaves their relative benefits unclear.

To address these limitations, we conduct the first thorough study of how to design market interfaces for the EV charging setting, and we make several novel contributions. First, we formalise the EV charging setting to capture several real-world challenges that give rise to complex preferences and we design two restricted interfaces for reporting preferences in this setting: one that reduces the dimensionality of the reporting space (but retains infinitely many options) and one that restricts the reporting space to a discrete set of options. To evaluate these interfaces with real users, we then develop a game that serves as an abstract representation of the EV charging setting. Using this game, we experimentally compare the restricted interfaces to a fully expressive interface in two large user

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

studies involving a total of over 300 participants hired on Amazon Mechanical Turk. We show that the restricted interfaces significantly alleviate the participants' cognitive burden (as measured by the time they deliberate). More surprisingly, they also lead to a significantly better performance than the fully expressive interface (up to 70% in some cases). Furthermore, we show that the choice of restriction can influence the energy consumption of participants without decreasing their utility. This indicates that market user interface design is an important topic to enable efficient future energy markets for end-consumers. Finally, we show that a reinforcement learning agent replicates general behaviour trends of human players, potentially paving the way for a new approach for optimising market interfaces without running large-scale user trials.

2. THE EV CHARGING PROBLEM

In this section, we present an abstract model of the EV charging problem. Our aim is to capture the key challenges that are inherent in the domain, while still retaining a succinct and general model. More specifically, we capture the following challenges that are found in realistic settings.

First, electricity has **no intrinsic value**. Its value instead depends on how it is utilised to complete journeys. Second, the problem is of an inherently **uncertain** nature. On one hand, this is a key feature of the market itself, as supply and demand may fluctuate significantly over time. On the other hand, there may be considerable uncertainty over what journeys the driver needs to complete in the future. Third, there are **complementarities**, i.e., the driver's preferences are typically highly nonlinear over the *quantity* of electricity they receive. For example, a driver may require a minimum overnight charge of 10 kWh to drive to work the next day. Receiving any less has no value.

Given this, we consider a general setting where EV drivers participate in a market mechanism to charge their vehicles. We abstract away from the particular market and assume that a driver simply reports her preferences for charging to an autonomous agent at regular intervals (e.g., once a day).[1] The agent then participates in the market on its owner's behalf and procures electricity for the vehicle while it is plugged in (e.g., overnight). At the end of the charging period, the driver can use the vehicle to complete journeys and derives value from doing so.

More formally, the problem consists of a sequence of n *days*, $D = \{1, 2, \ldots, n\}$. An EV starts with a given initial *state of charge (SOC)* $s_1 \in [0, s_{\max}]$ (in kWh). Then, at the start of each day d, the driver reports her charging preferences to an autonomous agent, which procures electricity from the market to charge the EV.

2.1 Electricity Market

We assume electricity is sold in discrete, unit-sized quantities (we use 1 kWh, w.l.o.g.). To participate in the market for day d, the EV driver reports her preferences for each quantity of electricity to her charging agent. This is done in the form of a *preference report vector* $w_d = [w_{d,1}, w_{d,2}, \ldots, w_{d,s_{\max}-s_d}]$, which indicates the driver's maximum willingness to pay for a given charge (up to the maximum capacity) on that day. This structure allows complementarities to be expressed — for example, a preference vector $w_d = [0, 4, 5]$ indicates that the driver is not willing to pay anything for receiving 1 kWh (e.g., because this is too little to complete any journeys), she will pay up to \$4 for 2 kWh and up to \$5 for 3 kWh.

Given this preference vector, the charging agent then participates in the market and obtains $x_d(w_d)$ units of electricity at a price of $p_d(w_d)$. These are uncertain, depending on the day's market conditions (e.g., supply of renewables and fluctuations in aggregate demand), but we assume that the agent maximises the driver's utility in expectation, such that truthful reporting is optimal, i.e., $\forall w_d, \hat{w}_d : \mathbb{E}[w_{d,x_d(w_d)} - p_d(w_d)] \geq \mathbb{E}[w_{d,x_d(\hat{w}_d)} - p_d(\hat{w}_d)]$, where w_d is the driver's true willingness to pay. In practice, this can be achieved by participating in an incentive compatible mechanism [13, 4] or by acting strategically on the owner's behalf, e.g., when optimising charging decisions given price predictions for real-time pricing [11]. Throughout the paper we will also assume that an agent's charging behaviour could influence prices on the same day, but has a negligible impact on the market conditions of the following days. This is a reasonable assumption in energy markets, where a single domestic consumer has little impact on future prices.

[1] For example market mechanisms, see [11, 13, 4].

2.2 EV Utilisation

Given the outcome of the market interaction, the new SOC of the EV is now $s'_d = \min(s_d + x_d(w_d), s_{\max})$. This can then be used by the EV driver to complete journeys. Specifically, every day, there is a set of *potentially available* journeys, $j_d \subseteq J$, where $J = \{1, 2, \ldots\}$ is the set of all journeys. These represent possible journeys the driver may wish to make during the day (e.g., driving to work or to the supermarket). To model realistic scenarios, at the time of reporting preferences, there may be uncertainty about which journeys will be available (e.g., because they depend on the weather or because some journeys will only be necessary in exceptional circumstances). To reflect this uncertainty, a journey is defined by a probability $r_j \in [0, 1]$ (we assume these are independent, although this can be easily relaxed, as discussed briefly in Appendix A) and a value $v_j \in \mathbb{R}$. The value of a journey reflects its importance to the driver and may represent the inconvenience cost incurred when missing the journey, or even the cost of alternative transport (e.g., to travel to work). The sets of potential journeys j_d are known in advance for all days, but the set of *actually available* journeys on day d, denoted by $j'_d \subseteq j_d$, only becomes known after charging is complete on that day.

Given the set of journeys j'_d, the driver now chooses which subset $a_d \subseteq j'_d$ of these to complete. Doing so reduces the EV's SOC, as given by an appropriate cost function $\gamma : 2^J \to \mathbb{R}_{\geq 0}$ (journeys that exceed the SOC cannot be completed). Furthermore, the driver receives the total value of these journeys. Thus, her total utility is the difference between the overall value derived from journeys over time horizon D, and the total costs incurred. In the following sections, we discuss two approaches for solving this optimisation problem computationally, one is optimal and one is a learning approach that may be a better model of human behaviour.

2.3 Optimal Solution

To optimise her utility, an EV driver needs to report a preference vector w_d on every day d that expresses the respective expected utility for receiving each possible quantity of charge. This is a complex problem, as it needs to take into account the (uncertain) availability of journeys not only on the current day, but also on future days, as well as likely future market conditions, as the battery of the EV allows the driver to store surplus electricity.

To derive the optimal strategy for a perfectly rational driver, we will assume that the driver has knowledge about the expected prices $p_d(w_d)$ and the distribution of allocations $x_d(w_d)$ for a given preference vector. With this, we can model the problem as a Markov Decision Process (MDP) [5], the full details of which are in Appendix A. The solution to such an MDP is a *policy* that selects both appropriate preference vectors to report to the charging agent and sets of journeys to complete, depending on the state (the current day d, the current SOC and the journeys available on a given day).

Solving this MDP optimally is NP-hard in general (as the journey selection generalises the Knapsack problem), but problems of realistic sizes can still be solved quickly. This is because we consider a limited time horizon here, allowing the use of backwards induction and dynamic programming. Furthermore, the possible journeys that a driver will seriously consider on a given day will be small, perhaps in the order of a dozen or fewer. We also discretise the state space of the SOC to include all reachable states, given the cost function γ. Finally, as the charging agent participates in the market optimally, the optimal preference vector is simply the driver's true valuation for each level of charge, and this is obtained readily from the MDP solution.

2.4 Reinforcement Learning Agent

Clearly, the optimal solution makes some assumptions that are unlikely to hold in practice (such as knowledge of the distributions of $p_d(w_d)$ and $x_d(w_d)$, and all potential future journeys). Hence, a second approach for solving the EV driver's decision problem is to design a reinforcement learning agent [19] (note this agent is different from the agent described above that interacts with the market). This approach does not require an explicit model of future market interactions or available journeys, but rather learns the optimal policy from repeatedly interacting with the environment and observing realised costs and rewards.

Specifically, we employ the widely-used Q-Learning algorithm [20] with an ϵ-greedy exploration strategy (see Appendix B for details). We hypothesise that this learning approach may be a good approximation of how real users interact with the system, by trying some preference reports, observing how the system responds and then making small adjustments to their strategies (rather than reasoning about the optimal strategy). In fact, reinforcement learning has been used as a computational model to explain learning and decision-making behaviours in animals [9].

3. RESTRICTED MARKET INTERFACES

In practice, the drivers solving the above optimisation problem will be humans, and, as we argued, they may not have the time or capacity to act optimally. Neither of the solution approaches discussed in the previous section are feasible for completely automating people's decision processes — the optimal solution requires full information about all possible future journeys, while reinforcement learning requires a long training phase until it performs well.

Instead, we here focus on simplifying the market interface for drivers. In particular, we use a range of interfaces that intentionally restrict the reporting space for the user (but without changing the underlying market mechanism). Knowing which interface to present to users can alleviate the cognitive burden, as the user has to consider fewer options.

Given this, we denote the full space of possible reports in a fully expressive interface by $W = \mathbb{R}_{\geq 0}^{s_{\max}}$. In the following, we will use two approaches for restricting this space by providing the user with an alternative set of reports, W', which maps to W through a function $\omega : W' \to W$, to determine the corresponding report that is used by the charging agent. Both approaches rely on significantly reducing the dimensionality of the decision space, while still retaining the ability for users to express a range of valuations.

3.1 Single Marginal Value with Quantity (SMV)

The first restriction, SMV, reduces the driver's possible reports to a single marginal value $m_d \in \mathbb{R}_{\geq 0}$ and a maximum quantity $q_d \in \mathbb{N}$ she wishes to acquire. Here, m_d expresses the value she gains for receiving each *additional* unit of electricity up to a total quantity of q_d. Thus, the space for this restriction is $W^{\text{SMV}} = \mathbb{R}_{\geq 0} \times \mathbb{N}$ and we denote a report by $w_d^{\text{SMV}} = (m_d, q_d)$. The corresponding mapping function is $\omega^{\text{SMV}}(m_d, q_d) = [m_d, 2m_d, 3m_d, \ldots, q_d m_d]$.

The rationale behind providing this restriction is that it reduces the dimensionality of the problem to two dimensions, which may be significantly easier for a user to understand and solve. However, the disadvantage of this approach is that the space of possible reports is still infinitely large. While q_d is discrete and restricted by the battery size, m_d may take on arbitrary values.

3.2 Finite Set of Alternatives

In the second restriction (FINITE), which has been used in related work [1, 17, 2], we select a finite subset of f alternatives from the full report space, $\{\alpha_1, \alpha_2, \ldots, \alpha_f\} \subset W$. The corresponding restricted report space is then $W^{\text{FINITE}} = \{1, 2, \ldots, f\}$, such that a report $w_i^{\text{FINITE}} = x$ expresses the user's choice of alternative α_x, i.e., $\omega^{\text{FINITE}}(x) = \alpha_x$.

The advantage of this restriction is that the driver has to consider a very small number of alternatives. In practice, f can be chosen to trade off the cognitive burden on the user with the expressivity of the space (typically, we envisage f to be a handful or less). The alternatives could be chosen to represent a cross section of the full report space, could be manually selected by domain experts or could even be selected by an autonomous agent that adjusts these alternatives to a particular user.

Note that both the optimal solution and the reinforcement learning approach can be adapted for the restricted interfaces through appropriate discretisation. The details are in Appendices A and B

4. THE BID2CHARGE TESTBED

To test how human participants interact with our interfaces, and to empirically determine which one works best, we designed a web-based game called Bid2Charge, which replicates the EV charging setting. We frame this as a game, as this is a low-cost, controlled way of gathering data from large numbers of users in a short time.

In more detail, in Bid2Charge, the player takes the role of an EV delivery van driver. This provides an intuitive explanation to players of what journeys represent (in the game, journeys are referred to as delivery tasks and result in a certain payment), what the objective of the game is (maximise overall profit) and what the uncertainty means (delivery tasks may or may not come up on a given day). We use an incentive-compatible auction based on the well-known VCG mechanism [10] as the market mechanism in this game, and so the player's reports in each of the market interfaces are framed as bids for this auction.[2]

Figure 1 shows the main game screen. At the top, there are some general statistics, showing the player's accumulated profit, the current day and current SOC. Below that, on the left, there is a task planning view. This provides the user with information about j_d, i.e., the tasks that are potentially available on the current day. Both the value, v_j, and the realisation probability, a_j, are shown for each task. Furthermore, the user can select subsets of tasks, $j' \subseteq j$ to inspect both the total value ($v(j')$) and the total cost ($\gamma(j')$) if those tasks are completed (here, $\gamma(j')$ is calculated based on the Euclidean distance of the shortest route past all tasks in j'). Note that interacting with this view does not affect the game — it simply provides the player with information about the available tasks. In Figure 1 the user has selected the \$10 and \$15 tasks and is informed that this will require 11 kWh in total for a reward of \$25.

The auction view, which is to the right of the task planning view,

[2]VCG fulfils the conditions for optimality in Section 2.1. It is dominant strategy incentive compatible for the current day (while assuming a probabilistic model of prices on future days).

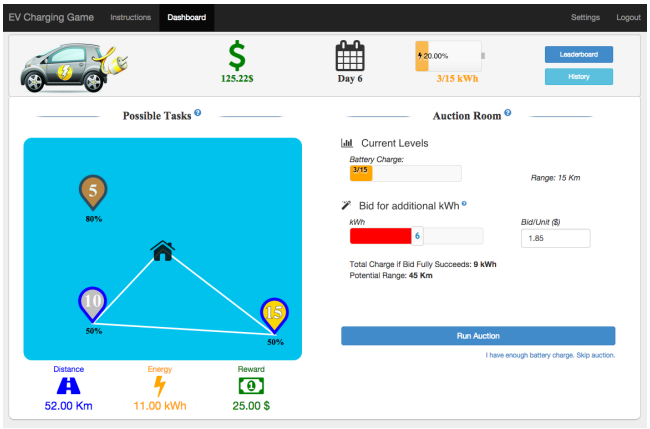


Figure 1: Main game view.

allows the player to submit their bid w_d for the current day. The view supports all three interfaces discussed in this paper, as shown in Figure 2: FINITE, SMV and a fully expressive interface. Only one of these is shown to each user. Here, the candidates for FINITE in Figure 2a can be customised by the game administrator, but this particular view shows $[1, 2, 3, \ldots]$, $[2, 4, 6, \ldots]$ and $[4, 8, 12, \ldots]$, which we also use for our experiments. The view for SMV in Figure 2b shows a user entering the bid $m_d = 1.5, q_d = 4$. Finally, the view for the fully expressive interface in Figure 2c shows a user entering $w_d = [0, 0, 2, 2, 5.5, 5.5, 5.5, 10.5]$. The user always has the option to skip the auction. On pressing the "Run Auction" button, they are informed of the outcome and then taken to a task view.

In the task view, representing the EV utilisation phase and shown in Figure 3, the user is presented with the realisation of tasks for the current day and can select their desired choice of tasks to complete. The options for this are given as a table with corresponding rewards and costs, and any dominated solutions are automatically removed.

5. EXPERIMENTAL EVALUATION

The purpose of our experiments is to investigate how real human users interact with the two restricted interfaces proposed in this paper, as well as the fully expressive interface, and whether the choice of a particular interface influences the performance of users. Our approach is to evaluate this through a randomised, controlled experiment, where we allocate market interfaces randomly to users, in order to exclude self-selection bias. Participants were given instructions only for the particular interface they were assigned to and were not told that others existed. In carrying out our experiments, we were guided by two main hypotheses:

Hypothesis 1: *Players using more expressive interfaces achieve the same or a higher profit than those using restricted interfaces.*

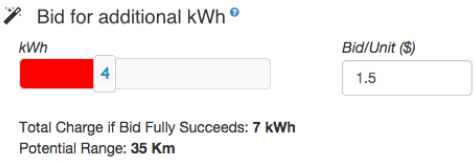
Hypothesis 2: *Players using more expressive interfaces spend more time on the game than those using more restricted interfaces.*

The first hypothesis is based on the fact that the more expressive interfaces offer more possible reports to the players. Experimental evidence also suggests that market user interfaces with more options lead to the same or better performance than those with fewer options [17]. However, more expressivity may also incur a higher cognitive burden, which is expressed by Hypothesis 2.

We tested these hypotheses through one initial experiment, where we asked participants to play through a single game consisting of 30 days. To further explore how players learn and improve over time, we carried out a second experiment, where we asked another set of participants to play through three identical 10-day games in

(a) FINITE:

(b) SMV:



(c) Fully-Expressive:

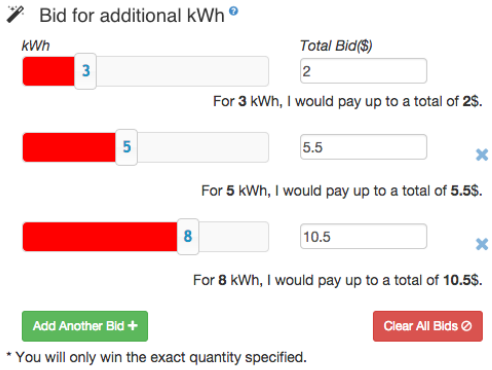


Figure 2: Auction Bidding Panels.

sequence (although journeys and their probabilities were identical, their realisations were not necessarily the same). In addition to validating the results from the first experiment, the main hypothesis we tested in the second setting was:

Hypothesis 3: *Players using more complex interfaces improve their profit more significantly over repeated plays of the game than those using more restricted interfaces.*

Furthermore, this more sequential setting allowed us to compare the human learning trends to those of the reinforcement learning agent we proposed in Section 2.4.

5.1 Player Recruitment

We recruited players from Amazon Mechanical Turk, a platform that allows requesters to advertise tasks to a large audience of online workers.[3] We asked workers to first read a set of instructions explaining the rules and objectives of the game, then they had to give consent to participating in a study as well as answer three basic questions about the rules (to test their understanding). We told workers the task would take about 25 minutes, they would receive a base payment of $2.50 and then a bonus based on the profit they made in the game. Specifically, in the first experiment, we offered $0.02 for each $1.00 earned in the game, while (due to budget constraints) in the second experiment we offered $0.01 for each $1.00 earned in each of the three games. In both experiments, this was capped at $3.00).[4] This bonus was chosen to ensure that the incentives of the player were aligned with the objective of the game. We assigned participants to interfaces using a block randomisation scheme, to achieve a balanced spread of participants across

[3] http://www.mturk.com/

[4] This maximum was set to limit our potential spend. Only three players managed to reach this.

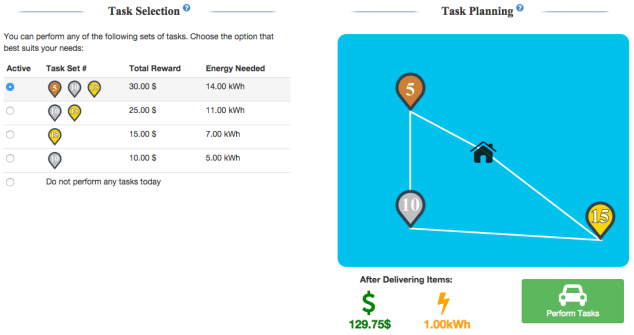


Figure 3: Task view with actually available tasks.

the interfaces. To ensure players understood the game, they had to correctly answer a short multiple-choice questionnaire. In the first experiment, 130 workers played the game; while in the second experiment 189 played three games each.

5.2 Experimental Parameters

To simulate the auction, we determine the marginal prices of units, denoted by $p_{d,x}$ for the xth unit, using random distributions. These are identical for all interfaces, on all days and for all participants (but the prices were sampled independently each time). Specifically, each price was determined by first setting $p_{d,0} = 0$, and then iteratively determining each $p_{d,x}$ as $p_{d,x} = p_{d,x-1} + \epsilon_x$, where ϵ_x was drawn from a uniform distribution $\mathcal{U}(0.2x - 0.2, 0.4x + 0.6)$. The rising mean of this distribution ensures that marginal prices for each unit generally increase. Given these prices, we then select $x_d(w_d) = \text{argmax}_x\, w_{d,x} - \sum_i^x p_{d,i}$.

The game in the first experiment was played for 30 simulated days, and we varied the number of tasks every 1–4 days (with between 1–6 tasks available every day). We did not give players information about tasks on future days (only the total number of days to play), and there was no prior information about the distribution of auction prices and allocation probabilities. This is reasonable because in real-world settings these would also be highly uncertain, and because we did not want to overwhelm players with a large amount of information. In the second game, a 10-day game was played three times by each player, enabling some learning.

5.3 Benchmarks

To establish upper and lower bounds for the possible performance of players, we compare them to a number of benchmarks: **Optimal** is the optimal strategy assuming a fully expressive interface. We also show two variants, **Optimal (SMV)** and **Optimal (FINITE)**, for the restricted interfaces. All of these are obtained by solving the MDP. **RandomGreedy** is a benchmark that places a random bid (chosen from the FINITE options) and then greedily chooses the highest-value tasks. Finally, **MaxGreedy** places a bid that is high enough to fully charge the EV each day and then greedily chooses the highest-value tasks. These two represent simple strategies that a worker could employ to complete the task with as little effort as possible (thus representing a lower bound on performance). Finally, we also show results for reinforcement learning strategies corresponding to the three interfaces, **QL(λ)**, **QL(SMV,λ)** and **QL(FINITE,λ)**, where λ is the number of episodes the game has been played by the agent. This performance is obtained by temporarily setting ϵ to 0.

5.4 Results of First Experiment

We first consider the overall performance in terms of the overall profit achieved, as this is the main objective of the game and to verify Hypothesis 1. Figure 4 shows this for the three interfaces with human participants (in red, plain), for the three optimal policies (in green, hatched), the two baseline benchmarks (in blue, finely hatched) and the reinforcement learning agent after 100 and 25000 iterations of the game, which are representative of early and late stages of learning (in orange, dotted). All results are shown with 95% confidence intervals. Focusing first on the performance of the optimal (in green), it is interesting to note here that there is little difference between the optimal performance in the restricted setting (in particular for SMV) and the fully expressive setting. This is encouraging, showing that, despite the severe restrictions, there is little loss in the utility players could, in theory, achieve.

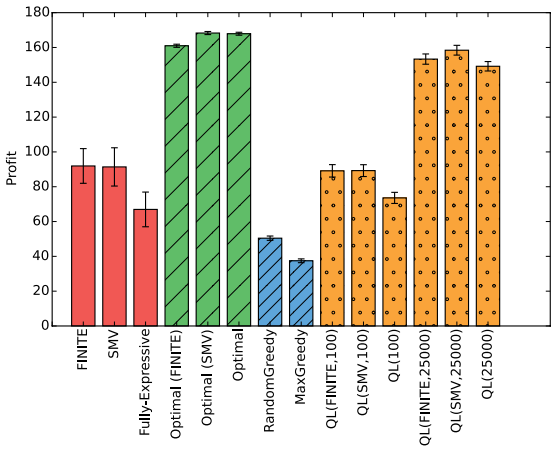


Figure 4: Average profit in first experiment.

Considering the performance of human players, these are generally situated between the optimal and the baseline benchmarks. Unsurprisingly, the humans perform significantly worse than a rational agent, given that the problem is highly complex due to its inherent stochasticity and combinatorial nature. However, when comparing the human players to the baseline benchmarks, there is a marked improvement. This provides some evidence that the participants are putting effort into the game rather than selecting strategies that require the least effort.

When comparing the performance of the human players using different market interfaces, several interesting trends emerge. First, we note that the choice of mechanism seems to have a significant influence on performance.[5] Counter-intuitively, the players using the fully expressive interface achieve the overall lowest performance with an average profit of only \$66.96, while players with SMV achieve an average profit of \$91.38 and players with FINITE achieve an average \$91.93.[6] This constitutes an improvement of over 35% compared to the fully expressive interface. This means our Hypothesis 1 needs to be rejected. One possible reason why the fully expressive interface does not perform as well is because of its substantial cognitive burden, because users are faced with a complex decision problem.

Considering the performance of the reinforcement learning agent, several interesting trends emerge. First, the fully-expressive interface consistently performs worse than the other two interfaces (with the same amount of learning). This is because the extremely large

[5]This is confirmed by ANOVA with $p = 0.001$.

[6]A post-hoc Bonferroni test confirms that there is a significant difference between the fully expressive mechanism and each of the other two (with $p = 0.004$). There is no significant difference between the performance of users with FINITE and SMV.

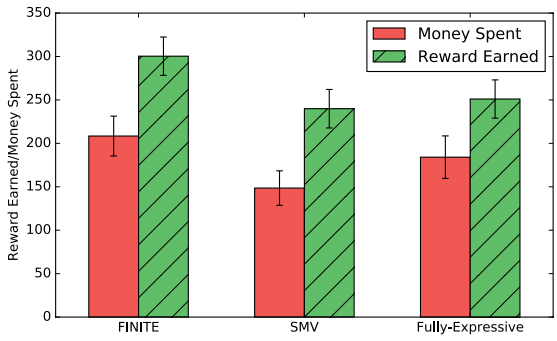


Figure 5: Average money spent and reward earned.

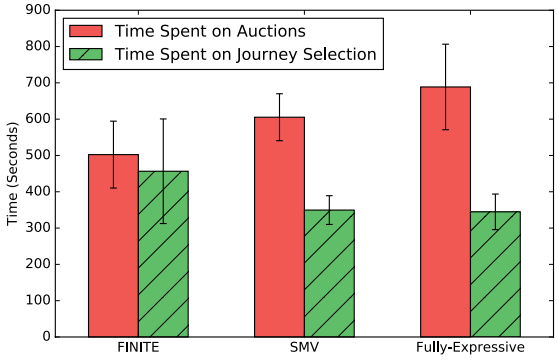


Figure 6: Average time spent by players on auctions and journeys.

reporting space takes longer to explore than in the other two interfaces. Furthermore, after 100 learning episodes, the performance of the reinforcement learning agent resembles that of human players, indicating that it may be useful for predicting human behaviour.

Although there is no significant difference in profit between FINITE and SMV for human players, interesting trends emerge when considering the actual money that was spent on acquiring electricity and how much was gained from completing tasks. This is shown in Figure 5. Here, players using FINITE spent an average $208.44 acquiring electricity and achieved an average reward of $300.37. In contrast, players using SMV spent only an average of $148.50 and earned $239.89. The differences here are highly significant.[7] Thus, while the profit for both is similar, FINITE induces a very different behaviour in the players — they spend significantly more on acquiring electricity and are then able to use this to complete a larger number of tasks. This is likely because SMV requires users to explicitly set a maximum number of units, thereby focusing them on this parameter and implicitly suggesting they restrict this demand. This is an interesting result, showing that significant changes in behaviour can be caused by the right choice of mechanism. Especially in the energy domain, low overall consumption may be a particularly desirable goal and SMV encourages this.

Next, to investigate Hypothesis 2, Figure 6 shows the time that players spent on the auction and task screens. Participants using the fully expressive interface spent the longest time on the auction (689 minutes on average), while participants using FINITE spent the least amount of time on the auction (502 minutes on average). This supports the hypothesis.[8]

[7]ANOVA confirms this for both metrics with $p \leq 0.002$. Post-hoc Bonferroni tests confirm differences between FINITE and SMV with $p = 0.001$.

[8]ANOVA ($p = 0.026$) and a Bonferroni test confirm a difference between the expressive and FINITE ($p = 0.021$.)

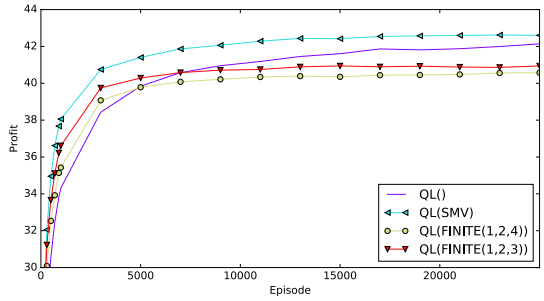


Figure 7: Average profit over number of learning episodes obtained by reinforcement learning agent.

5.5 Results of Second Experiment

To further examine whether our proposed reinforcement learning agent can be applied to optimising market user interfaces, we ran the agent in the setting of the second experiment and identified a small change to the FINITE strategy that led to an improvement in the agent's performance. Specifically, we decreased the marginal valuation reported in the third alternative from $4/kWh to $3/kWh. We used both alternatives in the second experiment, and in the following, we will refer to interfaces FINITE(1,2,3) and FINITE(1,2,4) to distinguish between these two. Figure 7 shows the performance of the reinforcement learning agent over time using the various interfaces.

Figure 8 shows the overall performance results from the second experiment, grouped by the three games each player completed. First, it is clear that there is a strong learning effect — on all treatments, the players perform better over time. On the first game, there is also a very clear performance difference between the interfaces. Fully expressive performs worse (average profit $15.6) while SMV performs best (average profit $26.8).[9] However, participants on the fully expressive interface then start to perform significantly better on subsequent games, supporting Hypothesis 3. This is likely because they begin to exploit the higher expressivity of the interface. At the same time, they are still taking significantly longer to deliberate during the auction phase (taking 373, 266 and 227 minutes for the three games, compared to 178, 123 and 116 of FINITE(1,2,3)), and they still do not outperform the restricted interfaces.

Finally, comparing the results with the predictions of the reinforcement learning agent, several trends are confirmed. First, FINITE(1,2,3) indeed outperforms FINITE(1,2,4), validating the agent's predictions and indicating that our approach of optimising the interface based on the agent's response is sensible. This is particularly promising, because there is no discernible difference in the *optimal* performance of the two interfaces (see Figure 8). Other trends indicated by the agent are also confirmed: SMV consistently performs well throughout, while the fully expressive interface initially performs poorly, but then catches up with the others as more learning and exploration take place.

6. CONCLUSIONS AND FUTURE WORK

As appropriate market mechanisms can efficiently allocate scarce resources to competing consumers, they hold considerable promise in addressing emerging challenges in the energy domain. However, in many realistic settings, including the EV charging problem,

[9]ANOVA ($p = 0.006$) and a Bonferroni test find a significant difference ($p = 0.036$) between these two.

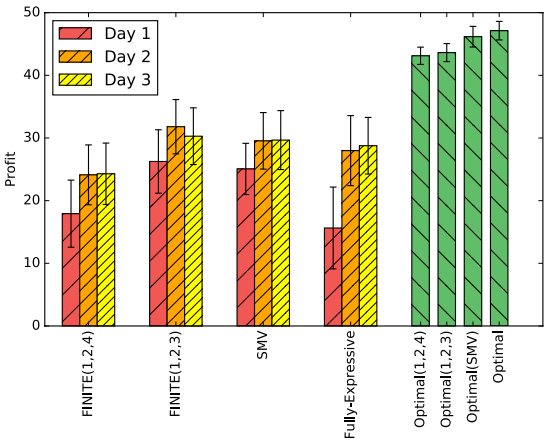


Figure 8: Average profit during each of the three games in the second experiment, along with benchmarks.

there are non-expert participants. In this paper, we studied how the cognitive burden on such participants can be alleviated by using restricted market interfaces.

This paper's extensive evaluation used a new framework called Bid2Charge that allowed us to compare people's behaviour in a simulated EV charging setting and using a variety of interfaces. As such, our work is the first comprehensive study of a wide spectrum of market interfaces, ranging from fully expressive to parameterised (SMV) to very limited (FINITE). In carrying out this study, we found that using restricted market interfaces has several key advantages over standard fully expressive approaches. First, participants spend less time deliberating, which indicates a lower cognitive burden. Second, these participants tend to perform better than those using a fully expressive, theoretically optimal interface. This performance gap is particularly pronounced during the first interactions, and then gradually closes with experience. Third, particular types of interfaces induce different behaviours in participants, while achieving the same utility. This could be a promising tool for nudging people towards particularly desirable behaviours, such as energy conservation. Last, we found that a reinforcement learning agent was able to predict broad trends in the relative performance of alternative interfaces. As such, we believe it constitutes a valuable tool for the evaluation and optimisation of market user interfaces.

In future work, we plan to build on this framework and develop new optimised interfaces that adapt to users. In particular, we will extend the reinforcement learning agent proposed in this paper to offer advice and support the user through autonomous decisions. Last, it is important to note that while Bid2Charge was evaluated using EV market mechanisms, it is a general framework which is easily adapted to other markets. Along these lines, we plan to study restricted interfaces and learning agents in other settings.

Acknowledgements

This work was supported by the EPSRC-funded ORCHID project and the Southampton Annual Adventures in Research grant.

APPENDIX

A. OPTIMAL SOLUTION

Here, we present the MDP formulation of the EV charging problem, describe an agent's policies and present the optimal solution.

A.1 MDP Formulation

We formalise the EV charging problem from Section 2 as a tuple $(\Sigma, \Sigma', J, j_{d\in D}, r_{j\in J}, v_{j\in J}, \gamma, X_{d\in D}, P_{d\in D})$, where:

- $\Sigma = D \times [0, s_{\max}]$ is the set of states *before* the market interaction, and $\Sigma' = D \times [0, s_{\max}] \times 2^J$ is the set of states *after* the market interaction (but before selecting from the available journeys). Both include the current day, d, and state of charge, s_d or s'_d, while a state $\sigma' \in \Sigma'$ also includes the set of available journeys j'_d.
- J, j_d, r_j, v_j and γ describe the journeys, their potential availability per day, realisation probabilities, values and journey costs, as defined in Section 2.
- $X_d : (W \times \{0, 1, \ldots, s_{\max}\}) \to [0, 1]$ is the probability distribution of x_d, i.e., $X_d(w_d, x)$ is the probability of obtaining x units when reporting w_d during a market interaction.
- $P_d : W \to \mathbb{R}$ is the expected price of a market interaction given a report, i.e., $P_d(w_d) = \mathbb{E}[p_d(w_d)]$.

The actions, transition probabilities and rewards of this MDP are fully described by the tuple above. Specifically, the available **actions** depend on the current state as follows. For $\sigma \in \Sigma$, the actions are the reports in W. For $\sigma' = (d, s'_d, j'_d) \in \Sigma'$, the set of actions is $\{a \in 2^{j'_d} \mid \gamma(a) \leq s'_d\}$, i.e., all subsets of the available journeys that can be completed with the current state of charge. **Transition probabilities** from a state σ to a state σ' are determined by the chosen report w_d and X_d (to determine s'_d) and by j_d and r_j (to determine j'_d). Transitions from a state σ' to σ are deterministically given by the chosen action a_d and cost function γ. **Rewards** are incurred when transitioning from one state to the next, and they correspond to the prices paid for electricity (negative) and the values derived from completing journeys (positive). Specifically, the former is given by $-P_d(w_d)$, while the latter is $\sum_{j\in a_d} v_j$.

A.2 Agent Policy

An agent's policy determines which preference vector to report for the market interaction, and which journeys to complete, given the current state. Thus, it is described by a tuple (π, π'), where $\pi : \Sigma \to W$ determines the reports and $\pi' : \Sigma' \to 2^J$ the journeys.

Given this, we can now define the expected utility of a policy (π, π') using a value function for each type of state. For state $\sigma = (d, s_d) \in \Sigma$:

$$V(d, s_d, \pi, \pi') = -P_d(\pi(\sigma)) + \sum_{i=0}^{s_{\max}} \Big[X_d(\pi(\sigma), i) \cdot \sum_{j'_d \subseteq j_d} R(j'_d)\, V'(d, \min(s_d + i, s_{\max}), j'_d, \pi, \pi') \Big], \quad (1)$$

where $R(j'_d) = \prod_{j\in j'_d} r_j \prod_{j\in j_d \setminus j_d} (1 - r_j)$ is the probability that j'_d is the set of journeys available.[10]

For state $\sigma' = (d, s'_d, j'_d) \in \Sigma'$, with $d < n$:

$$V'(d, s'_d, j'_d, \pi, \pi') = \Big[\sum_{j\in\pi'(\sigma')} v_j \Big] + V(d + 1, s'_d - \gamma(\pi'(\sigma')), \pi, \pi'), \quad (2)$$

while for the last day, $V'(n, s'_n, j'_n, \pi, \pi') = \sum_{j\in\pi'(\sigma)} v_j$.

[10]Note that while this definition of $R(j'_d)$ assumes independence between journeys, it could be redefined to account for correlated journey probabilities without changing the solution approach or its computational complexity.

Algorithm 1 Optimal Solution

```
1:  procedure FINDOPTIMAL
2:      Initialise π*, π'*, V and V' to be empty
3:      Initialise Γ to hold feasible charge levels given γ
4:      for d ∈ {n, n − 1, n − 2, . . . , 1} do
5:          for s ∈ Γ do
6:              for j'_d ⊆ j_d do
7:                  if d = n then
8:                      π'*(d, s, j'_d) =
                            argmax_{a⊆j'_d | γ(a)≤s} Σ_{j∈a} v_j
9:                  else
10:                     π'*(d, s, j'_d) =
                            argmax_{a⊆j'_d | γ(a)≤s} Σ_{j∈a} v_j
                                +V(d + 1, s − γ(a), π*, π'*)
11:             Calculate V'(d, s, j'_d, π*, π'*) using Equation 2
12:             Calculate π*(d, s, π*, π'*) using Equation 4
13:             Calculate V(d, s, π*, π'*) using Equation 1
14:     return (π*, π'*)
```

A.3 Optimal Solution

Given the above definition, the optimal policy (π^*, π'^*) is simply:

$$(\pi^*, \pi'^*) = \underset{(\pi,\pi')}{\text{argmax}}\, V(1, s_{\text{initial}}, \pi, \pi'), \tag{3}$$

where s_{initial} is the initial state of charge. As discussed in Section 2.3, this can be found using dynamic programming with backwards induction and by recognising that the agent's best strategy is to bid its true valuation for each level of electricity. This valuation is given directly by V', such that:

$$\pi^*(d, s_d) = \big[V'(d, s_d + 1, \pi^*, \pi'^*), V'(d, s_d + 2, \pi^*, \pi'^*), \ldots, V'(d, s_{\max}, \pi^*, \pi'^*)\big] \tag{4}$$

The full algorithm for computing the optimal policy is shown in Algorithm 1. In order to obtain optimal policies for the restricted interfaces, we can replace line 12 with the following:

```
13:  w* = argmax_{w∈W'} −P_d(ω(w)) + Σ_{i=0}^{s_max} [X_d(ω(w), i) ·
         Σ_{j'_d⊆j_d} R(j'_d) V'(d, min(s + i, s_max), j'_d, π, π')]
14:  π*(d, s) = ω(w*)
```

For SMV, this requires a discretisation of the marginal value. In our experiments, we use $m_d \in \{\$0, \$0.01, \$0.02, \ldots, \$5\}$, which is sufficiently fine-grained to lead to a near-optimal performance.

B. REINFORCEMENT LEARNING AGENT

Algorithm 2 shows the reinforcement learning algorithm we use as a more realistic benchmark than the optimal. This algorithm does not need knowledge of the underlying MDP and instead uses a Q-Learning approach to gradually learn the value of making particular reports in a given state (expressed using a $Q(\sigma, w)$ function). It takes three parameters: an exploration probability ϵ, a learning rate parameter α, and the set of possible reports W. In the experiments, we set $\epsilon = 0.2$ and $\alpha = 0.1$ (our results are not particularly sensitive to this choice). W is determined by the relevant interface.

In more detail, for each day d of each episode e, the agent first senses the current state σ (line 6). It then selects a report to submit using the PICKREPORT function. This function depends on the interface the agent is using. Algorithm 3 shows this for the Fully-Expressive interface, while Algorithm 4 is used for FINITE and SMV. Both approaches use an ϵ parameter that balances exploration (choosing a random, potentially untested report) with exploitation (choosing the best-performing report). As discussed in Section 2.4, the Fully-Expressive uses a local search technique to

Algorithm 2 Reinforcement Learning Agent

```
1:  procedure QLEARNINGAGENT(ε, α, W)
2:      Q̂ ← {}                                  ▷ State/action pairs that have been tried
3:      ∀σ ∈ Σ, w ∈ W : Q(σ, w) ← 0             ▷ Initialise Q-function
4:      for e ∈ {1, 2, . . .} do                 ▷ Episodes
5:          for d ∈ {1, 2, . . . , n} do
6:              σ ← SENSEAUCTIONSTATE()
7:              w ← PICKREPORT(ε, σ, Q, Q̂, W)
8:              REPORT(w)
9:              p_d ← OBSERVECOST()
10:             (d, s, j'_d) ← SENSEJOURNEYSTATE()
11:             if d = n then
12:                 a_d ← argmax_{a⊆j'_d | γ(a)≤s} Σ_{j∈a} v_j
13:             else
14:                 a_d ← argmax_{a⊆j'_d | γ(a)≤s} [ (Σ_{j∈a} v_j) +
                        argmax_{w∈W} Q((d + 1, s − γ(a)), w)]
15:             v ← Σ_{j∈a_d} v_j
16:             if (σ, w) ∉ Q̂ then
17:                 Q(σ, w) ← v − p_d
18:                 Q̂ ← Q̂ ∪ {(σ, w)}
19:             else
20:                 Q(σ, w) ← Q(σ, w) + α · (v − p_d − Q(σ, w))
21:             if RANDOM(0,1) < ε then
22:                 a_d ∈ {a ⊆ j'_d | γ(a) ≤ s}   ▷ Pick random journey set
23:             COMPLETEJOURNEYS(a_d)
```

generate new reports during exploration. Here, the MODIFYRANDOMELEMENT(w) function takes a report w, sets one randomly chosen element w_i to a random number (in the experiments, this is on the interval $[0, 6i]$), and then adjusts the other elements to ensure the vector is non-decreasing, by decreasing elements before i and increasing elements after i as necessary.

The agent then submits its chosen report, observes the incurred cost in the market and the new state (lines 8–10). It then picks the best set of journeys to complete, given the Q-values of states on the following day (lines 11–14). These state transitions are deterministic, given the costs of journeys, so we do not learn separate Q-values for journey choices. However, note that most Q-values for the next states will be zero initially, so the agent will start by greedily completing journeys to maximise its immediate reward.

Next, the agent updates the Q-function for its chosen report based on the overall profit achieved during the day (lines 15–20). Finally, with a small probability ϵ, the agent picks a new random set of journeys, which is again used for exploration; otherwise, it executes its chosen set a_d (lines 21–23).

Algorithm 3 Report Selection Function for Fully-Expressive

```
1:  procedure PICKREPORT(ε, σ, Q, Q̂, W)
2:      q ← {w | (σ, w) ∈ Q̂}
3:      if RANDOM(0,1) < ε ∨ |q| = 0 then
4:          if |q| = 0 then
5:              w ← [0, 0, . . . , 0]            ▷ s_max elements
6:          else
7:              w ∈ q                            ▷ Random report
8:          return MODIFYRANDOMELEMENT(w)
9:      else
10:         return argmax_w Q(σ, w)
```

Algorithm 4 Report Selection Function for FINITE and SMV

```
1:  procedure PICKREPORT(ε, σ, Q, Q̂, W)
2:      if RANDOM(0,1) < ε then
3:          return w ∈ W                         ▷ Random report
4:      else
5:          return argmax_w Q(σ, w)
```

REFERENCES

[1] D. Bergemann, J. Shen, Y. Xu, and E. Yeh. Multi-dimensional mechanism design with limited information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 162–178, New York, NY, USA, 2012. ACM.

[2] L. Blumrosen and M. Feldman. Mechanism design with a restricted action space. *Games and Economic Behavior*, 82(0):424 – 443, 2013.

[3] P. Dütting, F. Fischer, and D. C. Parkes. Simplicity-expressiveness tradeoffs in mechanism design. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 341–350, New York, NY, USA, 2011. ACM.

[4] K. Hayakawa, E. Gerding, S. Stein, and T. Shiga. Online mechanisms for charging electric vehicles in settings with varying marginal electricity costs. In *24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2610–2616, April 2015.

[5] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960.

[6] D. Kahneman. A psychological point of view: Violations of rational rules as a diagnostic of mental processes (commentary on stanovich and west). *Behavioral and Brain Sciences*, 23:681–683, 2000.

[7] K. Larson and T. Sandholm. Mechanism design and deliberative agents. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 650–656, 2005.

[8] P. Milgrom. Simplified mechanisms with an application to sponsored-search auctions. *Games and Economic Behavior*, 70(1):62 – 70, 2010. Special Issue In Honor of Ehud Kalai.

[9] P. R. Montague, S. E. Hyman, and J. D. Cohen. Computational roles for dopamine in behavioural control. *Nature*, 431(7010):760–767, 2004.

[10] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.

[11] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Agent-based homeostatic control for green energy in the smart grid. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):35, 2011.

[12] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence. *Commun. ACM*, 55(4):86–97, Apr. 2012.

[13] V. Robu, E. H. Gerding, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings. An online mechanism for multi-unit demand and its application to plug-in hybrid electric vehicle charging. *Journal of Artificial Intelligence Research*, 48:175–230, 2013.

[14] Royal Academy of Engineering. *Electric Vehicles: Charged with potential*. Royal Academy of Engineering, 2010.

[15] T. Sandholm and C. P. Boutilier. *Combinatorial Auctions*, chapter Preference elicitation in combinatorial auctions, pages 233–263. MIT Press, 2006.

[16] S. Seuken, K. Jain, D. S. Tan, and M. Czerwinski. Hidden markets: UI design for a P2P backup application. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 315–324, New York, NY, USA, 2010. ACM.

[17] S. Seuken, D. C. Parkes, E. Horvitz, K. Jain, M. Czerwinski, and D. Tan. Market user interface design. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 898–915, New York, NY, USA, 2012. ACM.

[18] H. A. Simon. Theories of bounded rationality. *Decision and organization: A volume in honor of Jacob Marschak*, pages 161–176, 1972.

[19] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press Cambridge, 1998.

[20] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.