Online Scheduling and Pricing for Electric Vehicle Charging

Mark M. Nejad, Lena Mashayekhy, Ratna Babu Chinnam, Daniel Grosu *

Abstract

We design strategy-proof online scheduling and pricing mechanisms for electric vehicle (EV) charging in a competitive environment. EV drivers submit their requests for charging services dynamically over time, and they can name their own price on the charging services. The mechanisms schedule EV charging and determine charging prices considering the incentives of both EV drivers and power providers. In addition, our proposed online mechanisms do not assume availability of information about future demand. Our charging mechanisms are preemption-aware allowing flexibility on when charging takes place. This is in alignment with power providers' load balancing goals. We perform extensive experiments to investigate the performance of our proposed mechanisms compared to that of the optimal offline mechanism. We analyze the various properties of our proposed mechanisms, in particular, we prove that they are strategy-proof, that is, truthful reporting of price and amount of charging is a dominant strategy for self-interested EV drivers.

Keywords: Electric vehicles, online charging, pricing, strategy-proofness, mechanism design.

1 Introduction

Electric vehicles promise to enable diversification of transportation energy feedstocks, reduce the dependency on fossil fuels, improve public health by lessening greenhouse gas emissions, and stimulate economic growth through the development of new technologies and industries. Widespread

*M. M. Nejad is with the School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK 73019.
L. Mashayekhy is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716.
R. B. Chinnam is with the Department of Industrial and Systems Engineering, Wayne State University, Detroit, MI 48202.
D. Grosu is with the Department of Computer Science, Wayne State University, Detroit, MI, 48202.
E-mail: mark.nejad@ou.edu, mlena@udel.edu, r_chinnam@wayne.edu, dgrosu@wayne.edu

adoption of electric vehicles is in alignment with sustainable transportation goals in their social, economic, and environmental aspects. Automotive companies are being challenged by environmentally conscious consumers and governments to produce affordable electric vehicles (Huang et al., 2013). Several companies from around the world have accepted the challenge, and more models of plug-in EVs (plug-in hybrid electric vehicles and pure battery electric vehicles) that can be charged from the electric grid are being introduced (Hybrid Vehicle Timeline, 2012). Unlike standard hybrid vehicles, plug-in hybrid EVs (PHEVs) also offer the ability to be recharged from an external electrical outlet. While pure battery electric vehicles (BEVs) currently offer limited driving range, PHEVs have an internal combustion engine besides an electric motor to overcome driving range issues.

Achieving large-scale adoption of EVs presents a number of challenges resulting from a current lack of supporting technologies/infrastructures and difficulties in overcoming technological barriers. Currently, EV drivers face long vehicle charging cycle times. In addition, they may also face long waiting times and uncertainty over availability of charging facilities. As EV usage for daily commute increases, enabling the ability to recharge these vehicles both in and away from base locations (e.g., residential locations) becomes more important. For example, some EV drivers may want to recharge their EVs at their destination locations such as workplaces, where their vehicles are parked for an extended duration. On the other hand, high electricity consumption of EVs is a major concern for electric utility companies making the load management of micro grids a challenge. EV consumption rates can be several times more than the average household consumption rate which can overload micro grids (Fairley, 2010). The impact on the grid is especially critical during peak grid demand hours. The existing electricity infrastructure may not be capable of providing the power to satisfy the surge in power demand under these situations.

While the utility companies will in the long-run work to address capacity shortages, they can significantly benefit from the development of scheduling and pricing mechanisms for EV charging that are cost effective while providing good services. They seek to deploy mechanisms that lead to balanced network load over time. One way to reach a better load balance is dynamic and preemption-aware scheduling. However, the problem of efficient scheduling and fair pricing of EV charging services is challenging, especially as both EV drivers and power providers can be seen as self-interested parties. EV drivers are interested in minimizing their costs and maximizing convenience, whereas utility companies would like to maximize their profits. When an EV is available for charging over an extended period (e.g., 8 AM to 4 PM), charging mechanisms can service that request (i.e., provide the charge) either in one continuous time slot, or in several discrete shorter time slots. A charging interruption may occur due to arrival of other urgent requests or the need for grid load balancing and necessitates preemption of scheduled requests.

Electric utility companies can also choose to sell their unallocated capacity in an auction platform. This is a win-win scenario for both providers and users, which allows providers to increase revenues while users can obtain charging units at lower prices. Such auction platforms will be of particular interest for PHEV users since they are not faced with range anxiety associated with pure BEVs (i.e., fear that a vehicle will run out of battery charge enroute). When the battery within a PHEV is depleted, the internal combustion engine works as a backup, providing a driving range comparable to conventional internal combustion engine vehicles. Moreover, EV drivers that are not in urgent need of charging but looking for bargains can also benefit from the auction-based platform. In the rest of this paper, we use the term EVs for electric vehicles whose users are interested in participating in this platform. EV users in urgent need for charging who are not willing to risk preemption can either bid high on the platform or use a conventional charging platform.

In this paper, we propose the first preemption-aware online mechanisms for scheduling and pricing EV charging in an auction-based platform. Users represent EV drivers whose charging requests arrive dynamically over time, at which point they name their own price (place their bids) to receive a certain amount of charging units by their departure. Our goal is to ensure that the micro grid capacity constraints are not exceeded, and those users who value the electricity the most are allocated and scheduled.

We consider a competitive environment where EV drivers compete for the limited supply of the electricity provider. These EV drivers are strategic users who are self-interested, meaning that they are interested in maximizing their own utility. Different users may have different time constraints and willingness to pay for charging services. Since users act strategically to maximize their own utility, they may misreport their preferences if it is in their best interest. Declaring lower bids than users' actual valuations may negatively affect other users and also lead to profit losses for the provider. Our goal is to design model-free mechanisms (i.e., we make no assumptions about future demand) that incentivises users to reveal their true preferences. Our proposed mechanisms consist of a scheduling algorithm and a dynamic pricing scheme for charging management of EVs considering their realtime demand.

1.1 Our Contribution

We introduce the problem of preemption-aware online scheduling and pricing (OSAP) for EV charging. The OSAP problem, given uncertainty about future arrivals, involves realtime scheduling and pricing of requests released over time (i.e., EVs that require a certain amount of charging by their departure) that share a scarce and perishable resource (i.e., electricity that is limited). We first propose an integer program to find the optimal schedule for the offline version of the problem, where all information about future supply and demand is known to the scheduler. We then propose an optimal offline mechanism using the proposed off-line scheduler and the VCG (Vickrey-Clarke-Groves) pricing scheme. In addition, we design a family of online mechanisms that solve the OSAP problem, where the requests arrive dynamically over time. The mechanisms are model-free, making no assumption about future demand, and they are invoked when a user places a new request or additional electricity capacity becomes available. We prove that all our proposed mechanisms are strategy-proof. This property incentivizes the EV users to report their preferences truthfully. We perform extensive experiments and show that our proposed online mechanisms are able to find near optimal solutions while satisfying the strategy-proofness property.

1.2 Related Work

Research on different decision problems related to EVs has attracted a great deal of attention in the past few years. Such research includes forecasting the EV market share (Glerum et al., 2014), designing energy-efficient routing of PHEVs (Nejad et al., 2016; Schneider et al., 2014), and proposing battery-swapping polices (Mak et al., 2013; Almuhtady et al., 2014). Kieckhäfer et al. (2014) proposed a hybrid simulation approach to estimate the evolution of EV market shares. Chocteau et al. (2011) investigated the impact of collaboration on the adoption of EVs among commercial fleets using concepts from cooperative game theory. Lin (2014) proposed a framework for optimizing the driving range by minimizing the sum of battery price, electricity cost, and range limitation cost as a measurement of range anxiety. Stüdli et al. (2014) studied the problem of returning electrical load to the grid, known as vehicle-to-grid, to reduce stress on the grid during peak times by injecting power back into the grid.

Automatic scheduling of EV charging has been studied from different points of view and considered different applications. Clement et al. (2009) proposed a coordinated charging scheduler in order to minimize the power losses and to maximize the grid load factor. Sundstrom and Binding (2012) proposed a load flow method for the problem of charging multiple EVs. However, strategic behavior of users (i.e., systematic manipulation of the system to gain unfair advantage) remains possible in their settings, where users misreport their preferences in order to receive preferential charging, leading to inefficient schedules that are not based on true users' requests. Gan et al. (2013) proposed a decentralized algorithm to optimally schedule EV charging by exploiting the elasticity of EV loads to fill the valleys in electric load profiles. Jin et al. (2013) investigated offline and online EV charging scheduling problems from a user's perspective by jointly considering the aggregator's revenue and users' demands and costs. Vasirani and Ossowski (2012) proposed a lottery-based solution for EV scheduling in order to ensure a level of fairness in the resulting scheduling in which a lottery system decides whether to charge a vehicle or not. However, none of these studies considered strategic users. In addition, they did not consider pricing.

Pricing EV charging is another line of research. Sioshansi (2012) investigates the incentives of EV drivers in making charging decisions with different electricity tariffs. In addition, he compares the cost and emissions impacts of these charging patterns to the ideal case of charging controlled by the system operator. Wei and Guan (2014) developed optimal electricity storage control policies to manage charging and discharging activities for PHEVs. Their proposed models capture the impact of the charging and discharging activities on real-time electricity prices. Flath et al. (2014) proposed a charging coordination model considering a spatial price component in order to analyze the loads from price-based EV fleet charging while at the same time accounting for distribution grid constraints. Misra et al. (2015) proposed a distributed dynamic pricing mechanism for the charging of PHEVs in a smart grid architecture. Once again, none of the above mentioned studies considered strategic users.

There is an extensive body of literature on mechanism design for scheduling that considers strategic users; the reader is referred to Heydenreich et al. (2007) for a survey. Mechanism design theory has been employed in designing strategy-proof mechanisms in several areas including spectrum auctions and cloud computing. In spectrum auctions, a government or a primary license holder sells the right to use a specific frequency band in a specific area using auction-based mechanisms (e.g., Zhou et al. (2008); Kasbekar and Sarkar (2010)). In cloud computing, a cloud provider offers computing services as commodities. Amazon Elastic Compute Cloud (Amazon EC2) offers auction-based cloud services through its spot instance, where users can bid on spare Amazon EC2 virtual machine instances. Several mechanisms have been designed for cloud auction markets (e.g., Mashayekhy et al. (2015b,a); Nejad et al. (2015)). Problems arising from each area have their own specific characteristics leading to fundamentally different problems. The characteristics of EV charging brings about new challenges in designing market mechanisms. Due to the unique combination of preemption-aware scheduling, strategy-proof pricing, multi-unit capacity and demand, multi-parameter requests, and limited information in our online setting, the existing mechanisms fail when applied to the EV charging problem. In this section, we will discuss the existing mechanisms, their settings, and limitations.

One of the open problems in mechanism design is designing optimal mechanisms in which the goal of the mechanism designer is profit maximization. This is a problem even for the case for just two items and two bidders in an auction (Sandholm and Likhodedov, 2015). Myerson (1981) proposed the design of strategy-proof revenue maximizing mechanisms for single item auctions in a single parameter setting. However, even for this simple case, the mechanism is not detail-free (i.e., it requires the seller to incorporate information about the bidders valuations and their distributions in the design of the mechanism). Hartline and Karlin (2007) discussed revenue-maximizing mechanism design in single parameter settings, where only one parameter of the user is private information. The problem of finding a revenue-maximizing combinatorial auction is NP-complete (Conitzer and Sandholm, 2004). The existence of detail-free revenue-maximizing auctions in general is unlikely (Conitzer and Sandholm, 2004; Chawla et al., 2013). In the absence of revenue maximizing auctions, several researchers resort to designing *randomized* optimal mechanisms (Celis et al., 2014; Dughmi and Roughgarden, 2014). Such randomized mechanisms are truthful in expectation, which is a weaker notion of truthfulness (strategy-proofness). The more common approach is designing strategy-proof mechanisms that implement social welfare maximization and yield high revenue. In this study, we consider multi-parameter settings, and our goal is to design strategy-proof mechanisms. In addition, to improve the revenue of the provider, we consider reserve prices. These reserve prices ensure the provider that the users bid at least that much in order to win. Our proposed mechanisms yield significantly higher revenue than the Vickrey-Clarke-Groves (VCG) mechanisms.

We focus on studies on online mechanism design, where users arrive dynamically over time. Online mechanism design is an important topic in the multi-agent and economics literature. The reader is referred to Parkes (2007) for an introduction to online mechanisms. One line of research in designing online mechanisms is to develop online variants of Vickrey-Clarke-Groves (VCG) mechanisms (Gershkov and Moldovanu, 2010; Parkes and Singh, 2003). These studies focus on Bayesian-Nash strategy-proofness. However, the focus of this paper is on the stronger concept of strategy-proof dominant-strategies. In addition, these studies are model based, and they rely on a model of future availability of supply and demand, while our proposed mechanisms are model-free. Hajiaghayi et al. (2005) and Porter (2004) considered model-free settings. Porter (2004) proposed a strategy-proof mechanism for online scheduling of jobs on a single machine. Hajiaghayi et al. (2005) studied the problem of online scheduling of a single, re-usable resource over a finite time period. They proved the strategy-proofness of their proposed mechanisms and derived lower bound competitive ratios. Lavi and Nisan (2004) considered multi-unit demand, and proposed an online auction model. In their model, however, the auctioneer must respond to each request immediately before considering other requests.

A number of studies have considered scheduling of EV charging with strategic users. Stein et al. (2012) proposed a model based online mechanism for pure electric vehicle charging. They introduced the use of pre-commitment in order to guarantee strategy-proofness of their proposed mechanism. In such a setting, when a mechanism precommits to a request, the request is neither preempted nor canceled. Gerding et al. (2011) proposed an online auction protocol for EV charging. In order to satisfy the strategy-proofness property, their mechanism allows burning of allocated electricity to some PHEVs. They showed that their proposed mechanism provides higher allocative efficiency than a fixed price system. Robu et al. (2012) proposed an online mechanism with strategic EV drivers allowing burning units. Robu et al. (2013) proposed an online mechanism for multi-unit demand and studied its application for charging PHEVs. They proposed two truthful allocation algorithms based on a greedy online assignment algorithm. Their allocation algorithms also allow occasional burning of allocated electricity to some PHEVs in order for their mechanisms to be strategy-proof. Gerding et al. (2013) proposed a two-sided online mechanism for advance reservations of charging, where EV users and providers can specify their preferences on time slots and number of units per time slot. Overall, our setting is more complex in comparison with the above-mentioned studies by jointly considering strategic users, allowing preemptions, and being model-free. The proposed preemption-aware scheduling and pricing mechanisms are also compatible with the load balancing objectives of utility providers. All these properties of the proposed mechanisms make them more compatible with the real-world settings.

2 Online Scheduling and Pricing Problem

In this section, we model the online scheduling and pricing (OSAP) problem for an electric utility provider that is providing charging service for EV users in a competitive environment. The utility provider is assumed to carry a limited electricity capacity C^t for EV charging during a discrete interval (of arbitrary choice) but the capacity might vary randomly from interval to interval by time of the day, $t \in \mathcal{T}$. Users compete for this limited supply while arriving dynamically over time at discrete intervals. User i requests l_i units of charge over a specified discrete interval $[a_i, d_i]$ and is willing to pay a maximum price of v_i if the service is completed on time. In this study, we consider that one unit of charging requires a unit of time, thus, users are requesting the charging units in terms of the amount of time that their EVs require to be charged. User i's bid (request) is denoted by $\beta_i = (a_i, l_i, d_i, v_i)$. For example, bid $(2, 1, 7, \$15)$ represents a user requesting 1 unit of charging, where the request arrives at time 2, expires at time 7, and her maximum price for the charging service is \$15.

The utility provider is able to (re)schedule the charging services for the different users at the arrival of any new user bid and/or change in available capacity.

We denote by $X_{N\times T}$ the charging schedule for all N users in the set $\mathcal{U}$ of users and T number of time intervals in the problem horizon, where x_i^t is 1 if user i's EV is scheduled for charging at time t, and 0, otherwise. Vector $X_i = (x_i^0, \ldots, x_i^T)$ represents the charging schedule for user i over time. Since the preemption of service is allowed, user i's charging might be completed over different intervals with interruptions. There have been some studies investigating the impact of different factors such as overcharging (too close to maximum battery capacity), temperature, and how frequently the charging services occur on the performance and lifespan of battery packs (Yilmaz and

Krein, 2013; Bashash et al., 2011; Cao et al., 2008). These factors can adversely affect the battery lifespan. Preemption may also have some negative impacts associated with battery longevity, which can translate into additional cost. To reduce such cost, the mechanisms can limit the number of preemptions occurring for each request. However, considering such factors that cause battery degradation is beyond the scope of this paper.

Each user i is characterized by a *valuation function* V_i defined as follows:

$$V_i(X_i) = \begin{cases} v_i & \text{if} \quad \sum_{t=a_i}^{d_i} x_i^t \geq l_i, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where X_i is the charging schedule of user i. We denote by W the *social welfare*, which is defined as the sum of users' valuations (i.e., the set of users with active service requests): $W = \sum_{i \in \mathcal{U}} V_i(X_i)$.

Given this setting, the problem of online scheduling and pricing of EV charging is to find a charging schedule and charging prices for users such that the total social welfare is maximized.

We denote by $\boldsymbol{\beta} = (\beta_1, \dots, \beta_N)$ the vector of requests of all N users, and by $\boldsymbol{\beta}_{-i}$ the vector of all requests except user i's request (i.e., $\boldsymbol{\beta}_{-i} = (\beta_1, \dots, \beta_{i-1}, \beta_{i+1}, \dots, \beta_N)$). We denote by Π_i user i's payment for receiving the charging service. We quantify user i's benefit through a *quasi-linear utility function* defined as the difference between the value she receives and the payment charged to her:

$$U_i(\boldsymbol{\beta}) = V_i(X_i) - \Pi_i \tag{2}$$

The users are self-interested, that is, they want to maximize their own utility. It may be beneficial for them to manipulate the service system and gain unfair advantage through untruthful reporting. A user can declare a higher value in the hope to increase the likelihood of obtaining her requested charging service. Strategic behaviors of such users may hinder other qualified users, leading to reduced revenue and reputation of the provider. With the increase in the number of EVs requiring charging, the potential for systematic manipulation will become a significant concern for utility providers. Our goal is to design strategy-proof mechanisms that solve the OSAP problem and discourage users from gaming the system through untruthful reporting.

The utility provider is also self-interested and wants to maximize its profit. In this setting, our goal is to give incentives to the utility provider to fulfill the entire request of a user rather than a partial allocation. In doing so, the utility provider receives payment from a user only if it provides her entire requested charging units. Note that in the absence of such setting, the utility provider can maximize its profit by greedily allocating charging units only to the users with the highest value per unit of charging at any time leading to the fractional OSAP problem (i.e., users are willing to pay for any fraction of their received request). Although, such strategy would result in higher profit for the utility provider, it does not consider the incentives of the users who want their entire requested charging units.

In an online setting, where complete information about future demand and supply is not available, designing an optimal mechanism is not possible. However, in an offline setting of the OSAP problem (SAP problem), we assume that such information is available, and thus, designing an optimal mechanism is possible. In the next section, we propose an optimal offline mechanism for the SAP problem that is used as a benchmark for evaluating the performance of our proposed online mechanisms.

3 Optimal Offline Mechanism

In this section, we propose an optimal offline strategy-proof mechanism for SAP, which considers that the information on all the future requests as well as supply is known *a priori*. A set $\mathcal{U}$ of N users submit their requests for the planning horizon of interest. We denote by $\hat{\beta}_i = (\hat{a}_i, \hat{l}_i, \hat{d}_i, \hat{v}_i)$ user i's declared request and valuation. Note that $\beta_i = (a_i, l_i, d_i, v_i)$ is user i's request and true valuation. Users are rational in the sense that they do not want to pay more than their valuation for their requests. A well-designed mechanism should incentivize users to participate. Such a property of a mechanism is called individual rationality and is defined as follows:

Definition 1 (Individual rationality)**.** *A mechanism is* individually-rational *if for every user i with true request β_i and the set of other requests $\boldsymbol{\beta}_{-i}$, we have $U_i(\beta_i, \boldsymbol{\beta}_{-i}) \geq 0$.*

In other words, a mechanism is individually-rational if a user can always achieve as much utility from participation as without participation. However, such mechanisms do not always incentivize users to report their requests truthfully. Our goal is to design a mechanism that is strategy-proof, i.e., a mechanism that incentivizes users to reveal their true requests.

Definition 2 (Strategy-proofness (Parkes, 2007)). *A mechanism is* strategy-proof *(or incentive compatible) if* $\forall i \in \mathcal{U}$ *with a true request declaration* β_i *and any other declaration* $\hat{\beta}_i$*, and* $\forall \hat{\boldsymbol{\beta}}_{-i}$*, we have that* $U_i(\beta_i, \hat{\boldsymbol{\beta}}_{-i}) \geq U_i(\hat{\beta}_i, \hat{\boldsymbol{\beta}}_{-i})$.

The strategy-proofness property implies that truthful reporting is a dominant strategy for the users. As a result, it never pays off for any user to deviate from reporting her true request, irrespective of the actions of the others.

Our first proposed strategy-proof mechanism is optimal, and is based on the Vickrey-Clarke-Groves (VCG) mechanism. An optimal schedule with VCG payments provides a strategy-proof mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973). We define our proposed optimal VCG-based mechanism for SAP as follows:

Definition 3 (VCG-SAP mechanism). *The VCG-SAP mechanism consists of a scheduling function* $\mathcal{S}$ *and a payment function* Π*, where*

i) $\mathcal{S}$ *is an optimal scheduling function maximizing the social welfare, such that* $X_i = \mathcal{S}_i(\hat{\boldsymbol{\beta}})$*, and*

ii) $\Pi_i(\hat{\boldsymbol{\beta}}) = \sum_{j \in \mathcal{U} \setminus \{i\}} V_j(\mathcal{S}_j(\hat{\boldsymbol{\beta}}_{-i})) - \sum_{j \in \mathcal{U} \setminus \{i\}} V_j(\mathcal{S}_j(\hat{\boldsymbol{\beta}}))$,

such that $\sum_{j \in \mathcal{U} \setminus \{i\}} V_j(\mathcal{S}_j(\hat{\boldsymbol{\beta}}_{-i}))$ *is the optimal social welfare obtained when user* i *is excluded from participation, and* $\sum_{j \in \mathcal{U} \setminus \{i\}} V_j(\mathcal{S}_j(\hat{\boldsymbol{\beta}}))$ *is the sum of all users' valuations in the optimal solution except user* i*'s value.*

Overall, we first identify the winning users and their optimal charging schedules. The prices are then determined based on the VCG pricing scheme.

In order to find the optimal scheduling function, we propose an Integer Program (IP) and define the decision variables over time $t \in \mathcal{T}$ as follows:

$$x_i^t = \begin{cases} 1 & \text{if a charging unit is allocated to user } i \text{ at } t, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$$y_i = \begin{cases} 1 & \text{if any charging unit is allocated to user } i, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In addition, we define indicator parameters as follows:

$$\delta_i^t = \begin{cases} 1 & \text{if } a_i \leq t \leq d_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

To maintain optimality, the solution should either fully service any particular request or not provide any service. The feasibility of the schedule to user i is indicated by δ_i^t. This indicator parameter ensures that the requested units are scheduled within time window $[a_i, d_i]$, if we choose to service the request.

The problem that needs to be solved to identify the winning bids and their optimal charging schedule can be formulated as an integer program (called SAP-IP), as follows:

$$\text{Maximize} \sum_{i \in \mathcal{U}} v_i \cdot [(\sum_{t \in \mathcal{T}} \delta_i^t x_i^t) - (l_i - 1) y_i] \tag{6}$$

Subject to:

$$\sum_{t \in \mathcal{T}} x_i^t \leq l_i, \forall i \in \mathcal{U} \tag{7}$$

$$\sum_{i \in \mathcal{U}} \delta_i^t x_i^t \leq C^t, \forall t \in \mathcal{T} \tag{8}$$

$$x_i^t \leq y_i, \forall i \in \mathcal{U}, \forall t \in \mathcal{T} \tag{9}$$

$$x_i^t = \{0, 1\}, \forall i \in \mathcal{U}, \forall t \in \mathcal{T} \tag{10}$$

$$y_i = \{0, 1\}, \forall i \in \mathcal{U} \tag{11}$$

$$\delta_i^t = \{0, 1\}, \forall i \in \mathcal{U}, \forall t \in \mathcal{T} \tag{12}$$

The objective function is to maximize the sum of all N users valuations. Only the values of the users who receive their complete charging requests are considered in the objective function. However, their allocation might be completed over different intervals (with interruptions) as long as they are within their requested time interval. Constraints (7) ensure that each user is serviced at most the requested amount. Constraints (8) guarantee that the allocation does not exceed the available capacity for any given time. Constraints (10) and (12) represent the integrality requirements for the decision variables and indicator parameters.

Once solved, SAP-IP finds the winning bids and their optimal charging schedule. Even though, the utility provider selects the same sets of users as winners using SAP-IP, it can incorporate different payment functions for pricing.

The utility provider can charge each winning user a price equal to her bid, meaning that, a winning user i will be charged $\Pi_i = v_i$. Such a mechanism is called first price mechanism. In this case, SAP-IP is a revenue maximization model since the objective is to maximize the total payment of the winning users. This is an upper bound on the revenue that the utility provider could extract from the users (Balcan et al., 2008). However, under such pricing, it is clear that users do not have incentives to bid their true values. If they bid their true values, they would obtain zero utility, $U_i = 0$. By bidding below their true values, they can potentially obtain positive utility. Therefore, the users need to strategize on how to bid and win, and they need to study the market and historical bids in order to lower their payments. As a result, by biding below their values, the provider experiences revenue losses. The drawback of the first price mechanism is that it is not strategy-proof, and the revenue of the provider is negatively impacted by users' strategic behavior.

On the other hand, the utility provider can adopt a different payment function in order to not put the burden of calculating and analyzing other users behaviors and bidding strategies on the users. According to the Revenue Equivalence Principle from auction theory, if the provider employs different mechanisms that select the same outcome (i.e., the same set of users as winners), these mechanisms lead to the same revenue in expectation (Nisan, 2007). In our case, the provider uses SAP-IP to determine the winning users and it charges them based on the VCG pricing scheme. As a result, following the Revenue Equivalence Principle, VCG-SAP provides the same expected revenue as the first price mechanism while guaranteeing strategyproofness. The charging prices are determined based on the VCG pricing scheme that also employs SAP-IP as a subroutine.

The execution time of VCG-SAP becomes prohibitive for large instances of the SAP problem. However, in an online setting, we do not have information about future bid requests or the capacity fluctuations, and thus, we resort to designing fast online mechanisms providing approximate solutions for the OSAP problem. Our goal is to design such online strategy-proof mechanisms that solve the OSAP problem effectively. The VCG-SAP mechanism will be used in our experiments purely as a benchmark for assessing the performance of the proposed online mechanisms.

4 Strategy-proof Online Mechanisms

In this section, we propose strategy-proof mechanisms (called MOSAP) for the OSAP problem. The goal of the mechanisms is to compute an efficient schedule even if $\hat{\beta}_i \neq \beta_i$ and calculate payments that incentivize users to report their true requests. We start by defining explicitly the required properties that our proposed mechanisms need to satisfy in order to guarantee strategy-proofness.

Definition 4 (Strategy-proof mechanism (Mu'Alem and Nisan, 2008))**.** *A mechanism is strategy-proof if it satisfies the following two properties:*

i) Monotonicity: the scheduling function $\mathcal{S}$ must be monotone, and

ii) Critical payment: the payment function Π *must be based on the critical payment.*

We define monotonicity in terms of the following preference relation $\succeq$ on the set of requests. A request $\hat{\beta}'_i = (\hat{a}'_i, \hat{l}'_i, \hat{d}'_i, \hat{v}'_i)$ is more preferred (i.e., $\hat{\beta}'_i \succeq \hat{\beta}_i$) if $\hat{a}'_i \leq \hat{a}_i$, $\hat{l}'_i \leq \hat{l}_i$, $\hat{d}'_i \geq \hat{d}_i$, and $\hat{v}'_i \geq \hat{v}_i$ for user i. That means the request $\hat{\beta}'_i$ is more preferred than $\hat{\beta}_i$ if user i requests fewer charging units, submits an earlier request, a later deadline, and a higher value. In our setting, for obvious reasons, users have no incentive to report an earlier arrival (i.e., $\hat{a}_i \leq a_i$) or a later deadline (i.e., $\hat{d}_i \geq d_i$) than their true arrival time and true deadline.

The monotonicity property indicates that any winning user who receives her requested charging units by declaring a request $\hat{\beta}_i$ will still be a winner if she requests a more preferred request. A user is a winner if her charging request is accepted and scheduled within her specified time interval. In the following, we describe the monotonicity property.

Definition 5 (Monotonicity)**.** *A scheduling function $\mathcal{S}$ is* monotone *if it selects user i with $\hat{\beta}_i$ as her declared request, then it also selects user i with a more preferred request $\hat{\beta}'_i$, i.e., $\hat{\beta}'_i \succeq \hat{\beta}_i$.*

In addition to a monotone scheduling function $\mathcal{S}$, any strategy-proof mechanism has a payment rule Π satisfying the critical payment property such that the payment of any user i, must be independent of her request. In the following, we describe the critical payment property.

Definition 6 (Critical payment)**.** *Let $\mathcal{S}$ be a monotone scheduling function, then for every $\hat{\beta}_i$, there exists a unique value v^c_i, called* critical payment*, such that $\forall \hat{\beta}'_i \succeq (\hat{a}_i, \hat{l}_i, \hat{d}_i, v^c_i)$, $\hat{\beta}'_i$ is a winning declaration, and $\forall \hat{\beta}'_i \prec (\hat{a}_i, \hat{l}_i, \hat{d}_i, v^c_i)$ is a losing declaration. $\Pi_i = v^c_i$ if user i wins, and $\Pi_i = 0$, otherwise.*

Algorithm 1 MOSAP-X(Event, X, Π)

$t \leftarrow$ Current time
$\mathcal{Q}^t \leftarrow \{\hat{\beta}_i | i \in \mathcal{U}$, i's request has not completed yet$\}$
$\mathcal{H}^t \leftarrow \{\hat{\beta}_i | i \in \mathcal{U}$, i's request can be completed at $t\}$
if $\mathcal{Q}^t = \emptyset$ or $C^t = 0$ **then**
  return
end if
$X^t \leftarrow$ MOSAP-X-SCH$(X, t, \mathcal{Q}^t, C^t)$
$X \leftarrow X \cup X^t$
$\Pi = \Pi \cup \{\hat{v}_i | x_i^t = 1\}$
$\Pi \leftarrow$ MOSAP-X-PAY$(t, \mathcal{H}^t, X^t, \Pi, C^t)$
return X, Π

In the following, we propose three different mechanisms for the OSAP problem. Since the three mechanisms are similar in structure, we present them as variants of a generic mechanism, called MOSAP-X, where X will be replaced with I, II and III to specify each of the three mechanisms. We define our proposed MOSAP-X mechanism as follows:

Definition 7 (MOSAP-X mechanism)**.** *The MOSAP-X mechanism consists of the scheduling algorithm MOSAP-X-SCH and the payment function MOSAP-X-PAY.*

The mechanisms in MOSAP-X prioritize users with different metrics such that in each mechanism the selection of the winning users and their schedule and payment might be different than those obtained by other mechanisms based on their given priority. MOSAP-I gives higher priority to users with higher values. However, MOSAP-II gives higher priority to users with higher value per unit of charge, while MOSAP-III determines the priority by taking into account both the value and the partial allocation.

MOSAP-X is an event handler, that is, it is invoked when a new user request arrives or available charging capacity changes. MOSAP-X finds the schedule of winning users by calling MOSAP-X-SCH and their payments by calling MOSAP-X-PAY. MOSAP-X is given in Algorithm 1. Our proposed mechanisms take as input an event, the current schedule set X, and the payment set Π.

MOSAP-X uses the following four variables defined as:

$\lambda_i^t = \sum_{\hat{a}_i \le \tau < t} x_i^\tau$; allocated amount to user i before time t

$\mathcal{Q}^t$: the set of feasible requests of the users that have not been scheduled completely yet (active requests). Formally, $\mathcal{Q}^t \leftarrow \{\hat{\beta}_i | i \in \mathcal{U}, [t \le \hat{d}_i] \wedge [\lambda_i^t < \hat{l}_i] \wedge [\hat{l}_i - \lambda_i^t \le \hat{d}_i - t]\}$, where $\wedge$ is the logical conjunctive operator.

Algorithm 2 MOSAP-X-SCH($X, t, \mathcal{Q}^t, C^t$)

1: $X^t \leftarrow \emptyset$
2: **for all** $i|\hat{\beta}_i \in \mathcal{Q}^t$ **do**
3: $\quad \lambda_i^t = \sum_{\hat{a}_i \leq \tau < t} x_i^\tau$
4: $\quad f_i = \hat{v}_i$, for MOSAP-I-SCH; or
$\quad f_i = \frac{\hat{v}_i}{\hat{l}_i}$, for MOSAP-II-SCH; or
$\quad f_i = \frac{(\lambda_i^t+1)\hat{v}_i}{\hat{l}_i}$, for MOSAP-III-SCH
5: **end for**
6: Sort all $\hat{\beta}_i \in \mathcal{Q}^t$ in non-increasing order of f_i
7: **for all** $\hat{\beta}_i \in \mathcal{Q}^t$ in non-increasing order of f_i **do**
8: $\quad$ **if** $C^t > 0$ **then**
9: $\qquad C^t = C^t - 1$
10: $\qquad x_i^t = 1$
11: $\quad$ **else**
12: $\qquad$ **break**;
13: $\quad$ **end if**
14: **end for**
15: **if** $C^t = 0$ **then**
16: $\quad$ **for all** $\hat{\beta}_i \in \mathcal{Q}^t$ for which $x_i^{t-1} = 1$ and $x_i^t = 0$ **do**
17: $\qquad$ Preempt user i's request
18: $\quad$ **end for**
19: **end if**
20: $X^t \leftarrow (x_0^t, \ldots, x_N^t)$
21: **Output:** X^t

$\mathcal{H}^t$: the set of requests that can be completed at time t. Formally, $\mathcal{H}^t \leftarrow \{\hat{\beta}_i | i \in \mathcal{U}, [t \leq \hat{d}_i] \wedge [\lambda_i^t < \hat{l}_i] \wedge [\lambda_i^t + 1 \geq \hat{l}_i]\}$

C^t: the available charging capacity at time t

Considering $\mathcal{Q}^t$, if the mechanism finds a better request than a current allocated request, it will preempt the allocated request with the intention of resuming its allocation at a later time. As a result, all active requests are in set $\mathcal{Q}^t$.

In lines 1 to 3, MOSAP-X sets the current time to t and initializes $\mathcal{Q}^t$ and $\mathcal{H}^t$. Then, it proceeds only if new resources and/or requests are available. MOSAP-X determines the scheduling by calling MOSAP-X-SCH. The scheduling function MOSAP-X-SCH returns X^t, the set of users who would receive their requested charging units at time t (line 7). The mechanism then updates the overall scheduling set X using the newly determined set X^t (line 8). Then, the mechanism initializes the payment of users in X^t by their submitted values (line 9). The mechanism updates the overall payment set Π by calling the payment function MOSAP-X-PAY (line 10). Finally, the mechanism returns the schedule and payment sets.

MOSAP-X-SCH is a dynamic scheduling function that considers the current capacity of the provider, and it finds the winning users in order to maximize the social welfare. MOSAP-X-SCH does not have prior information about the dynamics of users' arrivals, which is the case in online settings. However, it considers the feasibility of a schedule with $\mathcal{Q}^t$ based on the availability of the EVs for charging services. Our proposed scheduling algorithm MOSAP-X-SCH is given in Algorithm 2. We consider three algorithm variants for scheduling, MOSAP-I-SCH, MOSAP-II-SCH, and MOSAP-III-SCH. We define a metric called the *priority metric* for each algorithm.

MOSAP-X-SCH algorithm allocates the charging capacity to users in decreasing order of their priority metrics. We define the priority metrics of MOSAP-X-SCH as follows:

1) MOSAP-I-SCH: $f_i = \hat{v}_i$; 2) MOSAP-II-SCH: $f_i = \frac{\hat{v}_i}{\hat{l}_i}$; and 3) MOSAP-III-SCH: $f_i = \frac{(\lambda_i^t+1)\hat{v}_i}{\hat{l}_i}$. The priority metric for MOSAP-I-SCH gives higher priority to users with higher values. MOSAP-II-SCH considers the value per unit of charge as the priority metric. MOSAP-III-SCH gives higher priority to the users who have already received a partial allocation of their charging requests.

MOSAP-X-SCH sorts all requests in non-increasing order of priority metrics, f_i (line 6). Then the algorithm schedules the units requested by the sorted users in $\mathcal{Q}^t$ while resources last (lines 7-14). The mechanism uses this ordering for scheduling since the provider is interested in users who want to pay more. MOSAP-X-SCH tries to maximize the sum of the reported values of the users who get their charging units. By allowing preemption, MOSAP-X-SCH allocates charging units to users with higher priority while interrupting the allocation of users who are already allocated and have lower priority than the selected requests at the current time. The lower-priority request is suspended and is resumed as soon as possible (lines 15-19). Such a request is resumed when the value of its priority metric compared to those of other active requests is high enough to be selected. Since such a request has already received a part of the requested charging units, the mechanisms only need to provide the remaining units of the request in order to complete the request and receive the payment. Finally, MOSAP-X-SCH returns the set X^t of users who are scheduled at time t.

MOSAP-X-PAY finds the payment of each user by removing the user from the market and finding another schedule excluding that user. The critical payment of that user is calculated based on the bid of a user (i.e., user q) who wins the resources in the new schedule, but the user did not win them in the original schedule. In MOSAP-X-PAY, q represents the user who did not win in the current schedule, but could have won if user i had been excluded from the market. By doing

Algorithm 3 MOSAP-X-PAY($t, \mathcal{H}^t, X^t, C^t$)

```
1: W^t ← {β̂_i | i ∈ U, i's request is allocated and it is active at t}
2: for all β̂_i ∈ H^t ∪ W^t do
3:     λ_i^t = Σ_{â_i ≤ τ < t} x_i^τ
4:     f_i = v̂_i, for MOSAP-I-PAY; or
       f_i = v̂_i / l̂_i, for MOSAP-II-PAY; or
       f_i = (λ_i^t + 1) v̂_i / l̂_i, for MOSAP-III-PAY
5: end for
6: for all [[β̂_i ∈ H^t] ∧ [x_i^t = 1]] ∨ [β̂_i ∈ W^t] in non-increasing order of f_i do
7:     q = −1;
8:     X̄ ← MOSAP-X-SCH(t, H^t ∪ W^t \ β̂_i, C^t + 1)
9:     for all β̂_j | [x̄_j^t = 1] ∧ [x_j^t = 0] in non-increasing
               order of f_j, where f_j < f_i do
10:        q = j;
11:        break;
12:    end for
13:    if q ≥ 0 then
14:        Π_i ← min(Π_i, f_q), for MOSAP-I-PAY and MOSAP-III-PAY; or
           Π_i ← min(Π_i, f_q l̂_i), for MOSAP-II-PAY
15:    else
16:        Π_i ← r
17:    end if
18: end for
19: for all β̂_i | [λ_i^t < l̂_i] ∧ [t > d̂_i] do
20:    Π_i = 0
21: end for
22: Output: Π = (Π_1, ..., Π_N)
```

so, MOSAP-X-PAY bills each user i the minimum amount that she must declare to receive her request. Any user i's bid lower than this amount (i.e., v_i^c) would have led to the new schedule and selection of user q instead of user i while user i's request would have not been entertained.

The payment function MOSAP-X-PAY is given in Algorithm 3. This function calculates the *critical payment* of each user i if her EV is scheduled for charging at t. The critical payment of user i is the minimum value that she must report to receive the charging units at time t. MOSAP-X-PAY uses the set $\mathcal{H}^t$ of requests of users who are allocated or not allocated at t. This set does not include requests of users who are scheduled completely before t. MOSAP-X-PAY finds the set $\mathcal{W}^t$ containing the users who have been scheduled to receive their requests before t but their deadlines have not passed yet. Formally, $\mathcal{W}^t \leftarrow \{\hat{\beta}_i | i \in \mathcal{U}, [t \leq \hat{d}_i] \wedge [\lambda_i^t = \hat{l}_i]\}$. MOSAP-X-PAY calculates f_i for all users in $\mathcal{H}^t$ and $\mathcal{W}^t$ (lines 2-5). Then, MOSAP-X-PAY determines the payment for all users that have been scheduled at time t (i.e., $x_i^t = 1$) and will obtain their full requested charge by t. In addition, it updates the payment for all users that have been scheduled before time t who could have been scheduled at time t (i.e., users in $\mathcal{W}^t$). By doing so, MOSAP-X-PAY ensures that the critical payments of users are calculated. MOSAP-X-PAY calls the scheduling algorithm, MOSAP-X-SCH, without considering the participation of user i and with a capacity of $C^t + 1$ (i.e., the capacity before scheduling user i) (line 8). MOSAP-X-SCH returns the set of users $\bar{X}$ who would receive their requested charging at time t without user i's participation. Then, MOSAP-X-PAY tries to find a user j who had not been scheduled at t when user i participated (i.e., $x_j^t = 0$), and would have been scheduled at t if user i did not participate (i.e., $\bar{x}_j^t = 1$) (line 9). If MOSAP-X-PAY finds such a user, it stores her index q (line 10), and it determines the payment of user i based on the priority metric of user q (line 14); otherwise user i pays a reserve price $r \geq 0$ (line 16). In other words, the payment of user i is calculated based on the requests of losing users (i.e., that of user q), who would win if user i would not participate. This is the minimum value that needs to be reported by user i to obtain her request. Since the provider wants to guarantee a minimum revenue from each unit sold, the mechanism includes a reserve price. If this minimum price is set the same for all units and at all time points, this would not affect the properties of our proposed mechanisms. MOSAP-X-PAY keeps updating Π, and formally it calculates $\Pi_i = \arg\min_{\Pi_i^t \geq 0}[\hat{\beta}_i' = (\hat{a}_i, \hat{l}_i, \hat{d}_i, \Pi_i^t) \in \mathcal{S}_i(\hat{\beta}_i', \hat{\boldsymbol{\beta}}_{-i})]$. In addition, users who were not allocated any charging units pay zero (lines 19-21). Finally, the set Π is returned to the mechanism.

Under MOSAP-X, some of the users may not receive all their requested charging units. Even though these units are a few, MOSAP-X can adjust the allocation under well specified conditions. There are two possible ways to handle these partial allocations: burning and on-departure discharge. In burning, units are simply left allocated. For those allocated units, the provider does not receive any payment. In on-departure discharge, on departure of the user's EV, any allocated units are discharged from the battery. The model with on-departure discharge is more efficient in terms of resource utilization from the power provider's perspective, but it is not realistic to expect that we can discharge the partially allocated units from a car's battery on its departure. As a result, MOSAP-X uses burning in the case of partial allocation. The concept of burning has been used in the design of charging mechanisms in the literature (e.g., Robu et al. (2013)), and it is proven to be effective in terms of strategy-proofness of the mechanisms.

In addition, preemption allows our mechanisms to be flexible on when the charging takes place. Power providers can utilize such a feature of our mechanisms to shift some charging from peak grid

	$\hat{\beta}_i$	$\hat{a}_i$	$\hat{l}_i$	$\hat{d}_i$	$\hat{v}_i$
EV_1	$\hat{\beta}_1$	0	3	6	5
EV_2	$\hat{\beta}_2$	0	4	7	4
EV_3	$\hat{\beta}_3$	1	3	6	7
EV_4	$\hat{\beta}_4$	3	6	10	10
EV_5	$\hat{\beta}_5$	3	4	10	8

Table 1: User bids

	MOSAP-I f_i	MOSAP-II f_i	MOSAP-III f_i			
			$t = 0$	$t = 1$	$t = 2$	$t = 3$
EV_1	5	1.6	1.6	3.3	5.0	
EV_2	4	1.0	1.0	1.0	1.0	
EV_3	7	2.3		2.3	2.3	2.3
EV_4	10	1.6				1.6
EV_5	8	2.0				2.0
$\mathcal{S}$	$\{\hat{\beta}_4\}$	$\{\hat{\beta}_3, \hat{\beta}_5\}$	$\{\hat{\beta}_1, \hat{\beta}_3, \hat{\beta}_5\}$			
W	10	7+8 = 15	5+7+8=20			

Table 2: Execution of MOSAP-X

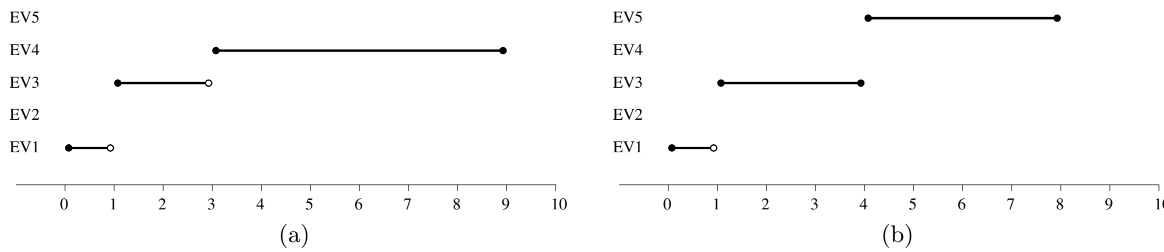


Figure 1: (a) MOSAP-I-SCH; (b) MOSAP-II-SCH.

demand hours to reduce stress on the grid during peak times.

Example 1. We show the execution of the mechanism by considering a setting with one unit of capacity available at each time slot and five users, denoted by EV_i, $i = 1, \ldots, 5$, as shown in Table 1. For example, user 1's bid $\hat{\beta}_1$ contains the following information: her request is submitted at time 0, with a deadline 6; she requests 3 units of charging with a bidding price 5. Table 2 show the execution of all three MOSAP-X-SCH mechanisms. In each column, the value of priority metrics, the set of winning users $\mathcal{S}$, and the obtained social welfare $\mathcal{W}$ are shown. For example, column f_i^I shows the priority metrics in MOSAP-I-SCH, the winning request is $\hat{\beta}_4$, and the obtained social welfare is 10. Figs 1-2 show the resulting schedules of the users obtained by the three mechanisms. Using MOSAP-I-SCH, EV_1 is selected at time 0, and then interrupted at time 1, when EV_3 is selected. At time 3, EV_4 is selected because of its highest priority, thus, EV_3 is interrupted. None of other users has higher priority than EV_4 until her EV receives all the requested charging units. At time 9, none of the users are active to receive a charging unit. This scheduling process by MOSAP-I-SCH is shown in Fig. 1.

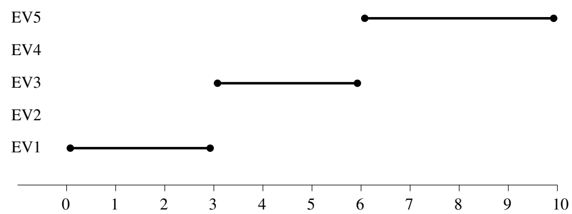


Figure 2: MOSAP-III-SCH

5 Properties of MOSAP

In this section, we investigate the properties of MOSAP-X. We first discuss the time complexity of the mechanisms, and then we show that the mechanisms are *individually rational* (i.e., truthful users will never incur a loss). We prove several lemmas in order to prove the strategy-proofness of MOSAP-X. At the end, we also present an example to analyze the effect of untruthful reporting on the users and the mechanisms.

The time complexity of MOSAP-X-SCH is $O(N \log N)$. This is because sorting the requests requires $O(N \log N)$, while checking the feasibility of the schedule for each user requires $O(1)$. The time complexity of MOSAP-X-PAY is $O(N^2 \log N)$ since it calls MOSAP-X-SCH that requires $O(N \log N)$ for each winning user. As a result, the time complexity of MOSAP-X mechanism is polynomial in the number of users.

Theorem 1. *MOSAP-X mechanisms are individually rational.*

Proof. We consider user i as a winning user. We need to prove that if user i reports her true request then her utility is non-negative. This can be easily seen from the structure of the MOSAP-X mechanisms. In line 14 of Algorithm 3, the payment for user i is set to $\Pi_i = f_q$ for MOSAP-I-PAY and MOSAP-III-PAY, and $\Pi_i = f_q l_i$ for MOSAP-II-PAY, where user q is the user who would have won if user i did not participate. Since user q appears after user i in the decreasing order of the priority metric in each of the selected mechanism, we have, $f_q \leq f_i$, thus, for each payment function, we have:

MOSAP-I-PAY: $\Pi_i \leq v_i$ because $v_q \leq v_i$;

MOSAP-II-PAY: Since $f_q \leq f_i$, we have $f_q l_i \leq f_i l_i$. Therefore, $\Pi_i \leq f_i l_i$. In addition, we have $f_i = \frac{v_i}{l_i}$. By substituting f_i, we have $\Pi_i \leq \frac{v_i}{l_i} l_i$, and thus $\Pi_i \leq v_i$;

MOSAP-III-PAY: In the last iteration of finding the priority metric to determine user i's payment, we have $\lambda_i^t = l_i - 1$. In addition, $f_i = \frac{(\lambda_i^t+1)v_i}{l_i}$. By replacing λ_i^t, we have $f_i = v_i$. In addition, $f_q \leq f_i$, thus we have $v_q \leq v_i$. As a result, $\Pi_i \leq v_i$.

MOSAP-X-PAY always computes a payment $\Pi_i \leq v_i$. As a result, the utility of user i (i.e., $U_i(\beta_i) = v_i - \Pi_i \geq 0$) is non-negative, and she never incurs a loss. In addition, a truthful user who does not win, is not incurring a loss since she obtains 0 utility. This proves the individual-rationality of MOSAP-X mechanisms. □

According to Definition 4, to obtain a strategy-proof mechanism, the scheduling function must be monotone, and the payment function must be based on the critical payment. We now prove the following lemmas and use them to prove that MOSAP-X mechanisms are strategy-proof. In order to prove that the mechanisms are strategy-proof, we need to prove that MOSAP-X-SCH is monotone, and MOSAP-X-PAY is based on the critical payment.

Lemma 1. *Let Γ_i be the space of possible requests user i may report to the MOSAP-X mechanisms. The scheduling algorithm MOSAP-X-SCH is monotone, for each $\hat{\beta}_i', \hat{\beta}_i \in \Gamma_i$, $\hat{\beta}_i' \succeq \hat{\beta}_i$, if user i wins by $\mathcal{S}(\hat{\beta}_i, \hat{\boldsymbol{\beta}}_{-i})$ then she wins by $\mathcal{S}(\hat{\beta}_i', \hat{\boldsymbol{\beta}}_{-i})$. In other words, if user i wins by bidding $\hat{\beta}_i$, then she will also win if she reports a more preferable bid $\hat{\beta}_i'$.*

Proof. Request $\hat{\beta}_i'$ is more preferred than $\hat{\beta}_i$ if user i requests fewer charging units, submits an earlier request, a later deadline, and a higher value. It is only beneficial for the user to misreport $\hat{a}_i \geq a_i$ and $\hat{d}_i \leq d_i$. These cases of misreports do not represent more preferable bids, and thus, we will focus on misreports of v_i and l_i.

If user i reports $\hat{v}_i' \geq \hat{v}_i$, her priority metric increases in all the MOSAP-X mechanisms (i.e., $f_i' \geq f_i$). As a result, bid $\hat{\beta}_i'$ will be selected as a winner by the MOSAP-X mechanisms if $\hat{\beta}_i$ is also selected as a winner.

Similarly, if a user is selected as a winner when reporting $\hat{l}_i$, she will also be selected when reporting $\hat{l}_i' \leq \hat{l}_i$. This is due to the fact that her priority metric either increases in case of MOSAP-II-SCH (i.e., $f_i' \geq f_i$), or remains the same in the cases of MOSAP-I-SCH and MOSAP-III-SCH (i.e., $f_i' = f_i$). In either cases, user i will be selected as a winner by the MOSAP-X mechanisms. □

Lemma 2. *The payment function implemented by MOSAP-X-PAY is based on the critical payment.*

Proof. We need to show that Π_i determined by MOSAP-X-PAY is the minimum value that user i must report to get the complete charging service. User i's payment is $\Pi_i = f_q$ for MOSAP-I-PAY and MOSAP-III-PAY, and $\Pi_i = f_q\hat{l}_i$ for MOSAP-II-PAY (line 14), where q is the index of user q appearing after user i based on the non-increasing order of the priority metrics (line 4), and she would have won if user i did not participate.

If user i submits a lower value $\hat{v}'_i < \Pi_i$, user i's new priority metrics are decreased. We consider the following cases:

MOSAP-I-PAY: $f'_i = \hat{v}'_i$. Since $\hat{v}'_i < \Pi_i$ and $\Pi_i = f_q$, we have $f'_i < f_q$

MOSAP-II-PAY: $f'_i = \frac{\hat{v}'_i}{\hat{l}_i}$. Since $\hat{v}'_i < \Pi_i$, $f'_i < \frac{\Pi_i}{\hat{l}_i}$. Since $\Pi_i = f_q\hat{l}_i$, we have $f'_i < \frac{f_q \cdot \hat{l}_i}{\hat{l}_i}$. Thus, $f'_i < f_q$

MOSAP-III-PAY: $f'_i = \hat{v}'_i$. Since $\hat{v}'_i < \Pi_i$ and $\Pi_i = f_q$, we have $f'_i < f_q$

As a result, we have $f'_i < f_q$ in all cases. This means that by submitting this lower bid, user i will appear after user q, who did not win. MOSAP-X-PAY will select user q instead of user i. As a result, if user i reports a bid below the minimum value (i.e., Π_i), she loses; otherwise, she wins. This unique value is the critical payment for user i. This, together with the fact that losing users pay zero, prove that the payment function implemented by MOSAP-X-PAY is the critical payment. □

Theorem 2. *MOSAP-X mechanisms are strategy-proof.*

Proof. Lemma 1 proves that the MOSAP-X-SCH is monotone. Lemma 2 proves that the MOSAP-X-PAY implements the critical payment. It follows from Parkes (2007) that MOSAP-X are strategy-proof. □

MOSAP-X-SCH is monotone since when a user submits a more preferred bid, her chances of winning do not decrease. For our monotone scheduling function, MOSAP-X-PAY satisfies the critical payment such that for every user, there is a critical value in which the user switches from winning to losing. In addition, MOSAP-X-PAY determines the price of each user independent of her bid. In doing so, the mechanism excludes that user and calculates the critical payment. Therefore, strategic behavior of that user cannot affect the outcome of the mechanism.

We show that our proposed mechanisms are robust against manipulation by users through the following example. To analyze the effect of untruthful reporting on the utility of the users participating in the MOSAP-II mechanism, we consider three users EV_1, EV_2 and EV_3, whose true

	β_i	$\hat{a}_i$	$\hat{l}_i$	$\hat{d}_i$	$\hat{v}_i$
EV_1	β_1	0	3	6	5
EV_2	β_2	1	3	6	6
EV_3	β_3	2	2	4	4

Table 3: Users' true requests

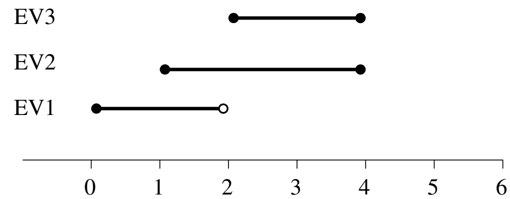


Figure 3: Example MOSAP-II-SCH

Table 4: Different scenarios for user EV_3's request declaration

Case	$\hat{\beta}_3$	Scenario	Status	Payment	Utility
I	$< 2, 2, 4, 4 >$	$\hat{v}_3 = v_3$	W	3.3	0.7
II	$< 2, 2, 4, 5 >$	$\hat{v}_3 > v_3$	W	3.3	0.7
III	$< 2, 2, 4, 3.5 >$	$\hat{v}_3 < v_3$	W	3.3	0.7
IV	$< 2, 2, 4, 3 >$	$\hat{v}_3 < v_3$	L	0	0
V	$< 2, 3, 4, 4 >$	$\hat{l}_3 > l_3$	L	0	0
VI	$< 2, 1, 4, 4 >$	$\hat{l}_3 < l_3$	W	3.3	0.7
VII	$< 3, 2, 4, 4 >$	$\hat{a}_3 > a_3$	L	0	0
VIII	$< 2, 2, 3, 4 >$	$\hat{d}_3 < d_3$	L	0	0

requests are shown in Table 3. We consider the electricity capacity of $C = 2$ units. MOSAP-II-SCH schedules these users as shown in Fig 3, where all users declare their true requests. User EV_2 and EV_3 are selected as winners, and the payments of the winning users based on MOSAP-II-PAY are 5 and 3.3, respectively.

We assume that user EV_3 reports a different request, $\hat{\beta}_3$, from her true request $\beta_3 =< 2, 2, 4, 4 >$. As shown in Table 4, we analyze different scenarios, where user EV_3 submits different requests. In addition, we present the payment and utility of the user for all the cases.

In case I, user EV_3 submits her true request, that is, $\beta_3 = \hat{\beta}_3$. In this case, user EV_3 wins, and receives the requested charging units. The mechanism charges her \$3.3, and her utility is 4-3.3=0.7. In case II, user EV_3 submits a request with a higher bid $\hat{v}_3 = 5$. In this case, user EV_3 is still a winner and the mechanism determines the same payment for her as in case I, leading to a utility of 0.7. In case III, she submits a request with a lower bid $\hat{v}_3 = 3.5$, which is not less than the price determined by our mechanism (i.e., \$3.3). Thus, user EV_3 is still winning, and the mechanism charges her the same amount as in case I. However, if user EV_3 submits a request with a bid below the critical payment, she will not obtain her requested charging units, leading to zero utility. This is shown in case IV, where user EV_3 submits a bid $\hat{v}_3 = 3$. We now investigate scenarios in which user EV_3 requests a different amount of charging units than her true request. In case V, she requests more units of charging, $\hat{l}_3 = 3$, instead of 2 units in the case of her true request, case I. In this case, user EV_3 is not selected, leading to zero utility. In case VI, the user requests fewer charging units. In this case, user EV_3 is still a winner and the mechanism determines the same payment for her as in case I. This is due to the fact that the user declared a more preferable request than her actual request. The user does not gain more utility by such declarations. In case VII, the user submits her request with a later arrival, which makes the allocation unfeasible. In case VIII, user EV_3 submits her request with an earlier deadline, which makes the allocation unfeasible leading to zero utility for the user. We showed that if a user submits a request untruthfully, she can not increase her utility.

6 Experimental Results

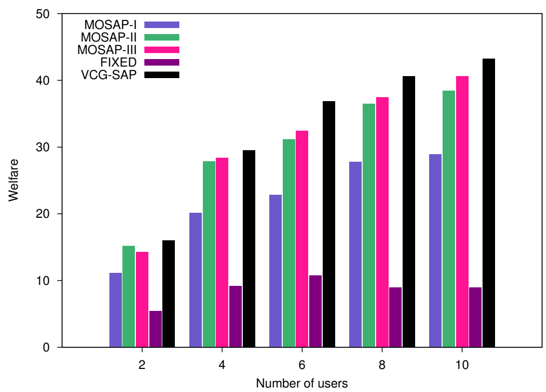
We perform extensive experiments in order to investigate the properties of the proposed mechanisms, MOSAP-X. We compare the performance of MOSAP-X with that of VCG-SAP and FIXED, where VCG-SAP solves optimally the offline version of the problem, and FIXED is a fixed-price mechanism. In the FIXED mechanism, each unit of charging is allocated to a user chosen randomly. If a user receives her total requested units in the FIXED mechanism, she pays the reserve price. We rely on the VCG-SAP and FIXED results as benchmarks for our experiments. All algorithms are implemented in C++. SAP-IP is implemented using APIs provided by IBM ILOG CPLEX Optimization Studio Multiplatform Multilingual eAssembly. In this section, we describe the experimental setup and analyze the experimental results.

6.1 Experimental Setup

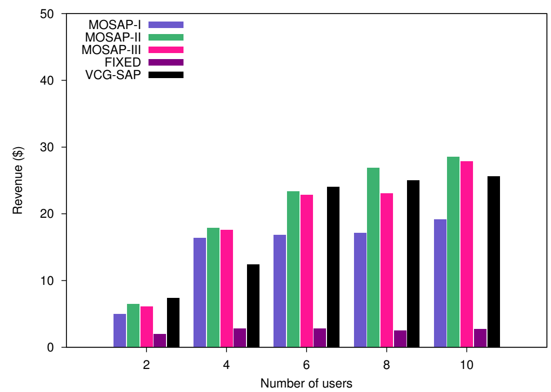
We design the experimental setup based on publicly available data on an EV in production (Tesla Model S, 2015). Following Robu et al. (2013), we consider a general synthetic setting, in which we generate users and their requests from simple distributions. The main reason for this setup is to generate results that are easily reproducible. Our experiments run with a variable number of EV charging requests over a 24-hour period. For each user i, we sample the EV arrival time a_i from the discrete uniform distribution on $\{0, 1, 2, \ldots, 23\}$ and the EV departure time from $\{a_i, a_i+1, \ldots, 23\}$. We assume that electricity is allocated in hourly time slots, where each unit corresponds to 10 kWh which is the approximate energy obtained through a Tesla Model S Single Charger for an hour of charging (Teslamotors, 2015). We sample the number of required units l_i uniformly at random from $\{1, 2, \dots, 5\}$, where 5 units of 10 kWh represent the maximum of 50 kWh charging requests. This amount is about 70% of the Model S maximum battery capacity (i.e., 70 kWh), which adds 150 miles to the range of the EV. The reason we choose upto 70% of the capacity (50 kWh out of 70 kWh) as the maximum amount of charging requests is that EV drivers participating in our proposed market are less sensitive to not receiving their requests than the EV drivers in the fixed price market who may need charging services immediately. Finally, we generate v_i from $f(x) = 10e^{-x}$, where e^{-x} is an exponential distribution with rate 1. We note that one hour to five hours of charging in a fixed price market costs between \$1.2 and \$6 based on the US national average of \$0.12 per kWh. However, in an auction setting, users bid in a wider range depending on their preferences. In our setting, a few users bid higher than the fixed price market to ensure receiving their charging requests (and without preemption) while they will be charged less than their bids based on the market demand. Most users, however, bid bellow the fixed price to reduce their charging cost. In addition, we consider 0.5 as the reserve price to guarantee a minimum profit for the utility provider.

6.2 Analysis of Results

We analyze three sets of experiments: small-scale, large-scale, and sensitivity analysis on capacity. We compare the performance of MOSAP-X, VCG-SAP, and FIXED for different number of users and amount of capacity. In the small-scale and large-scale setups, we fix the amount of available capacity while varying the number of arriving users. In the sensitivity analysis, we fix the number of users while we analyze effects of changes in the amount of available capacity on the performance of both mechanisms. We will show that VCG-SAP cannot find the optimal solution in feasible time for all instances of the SAP problem in the large-scale and sensitivity analysis cases. In order to have a comparison for all instances of the SAP problem, we present the results of the small-scale experiments, where VCG-SAP is able to find the optimal solution for all instances. We record the welfare, the revenue, the execution time, the total number of served users, and the total allocated units with payment for each mechanism.



(a)



(b)

Figure 4: Small-scale experiments with 1 unit capacity: (a) Welfare; (b) Revenue.

6.2.1 Small-scale experiments

We analyze the performance of MOSAP-X, VCG-SAP, and FIXED, where the available capacity is 1 unit of 10 kWh. In this case, the number of users that arrive every hour is between 2 and 10. Fig. 4a shows the welfare obtained by the mechanisms. These results show that MOSAP-II and MOSAP-III obtain a welfare very close to that obtained by the optimal VCG-SAP mechanism. Such results are very promising given the fact that MOSAP-X is an online mechanism which does not have any information about future demand. However, VCG-SAP is an offline mechanism and has all the information available a priori. However, the welfare obtained by MOSAP-I is not close to the optimal results because it does not consider the amount of requested charging units by users in its scheduling function. As expected, since FIXED randomly allocates the unit to users, its obtained welfare is far from the optimal results. Each obtained welfare by the mechanisms is an upper bound on the revenue of the utility provider in that case. This is due to the fact that if the provider employs the first price mechanism ($\Pi_i = v_i$ for winning user i), the welfare is equivalent to the revenue of the provider. For example, the maximum revenue that MOSAP-I can obtain with two users per hour is \$11.12.

Fig. 4b shows the revenue achieved by the provider when using the mechanisms. Note that the VCG-SAP is optimal in terms of welfare and not the revenue. The results show that MOSAP-II obtains the highest revenue among all the mechanisms. By employing our proposed payment functions, the mechanisms obtain high revenue for the provider. Note that if the provider employs

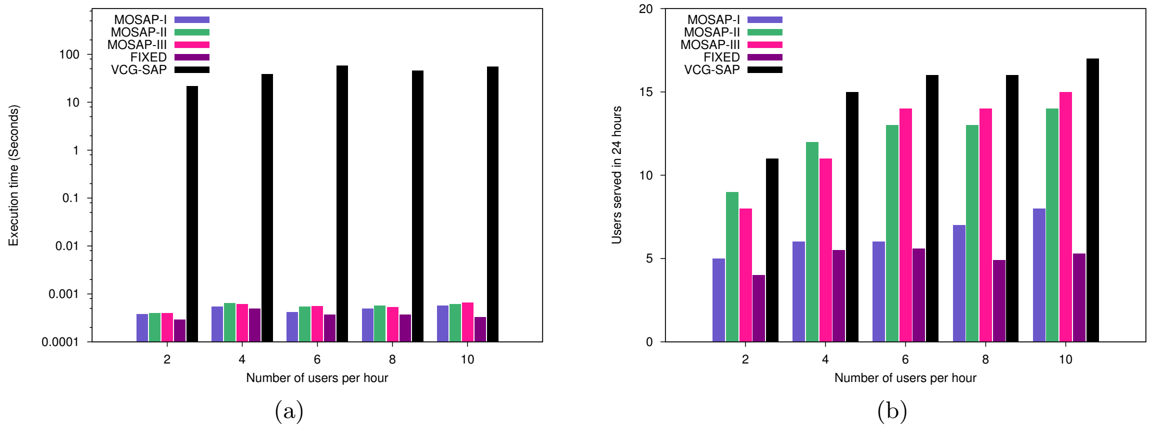


Figure 5: Small-scale experiments with 1 unit capacity: (a) Execution time; (b) Total served users.

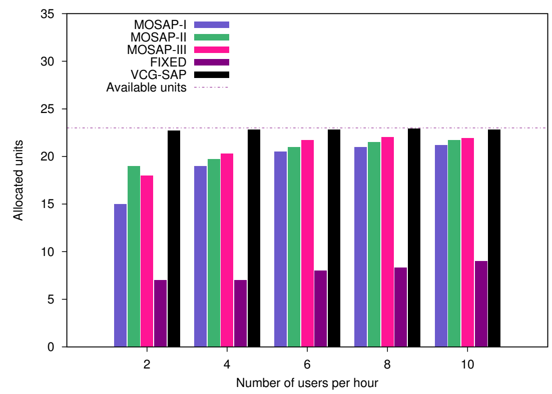
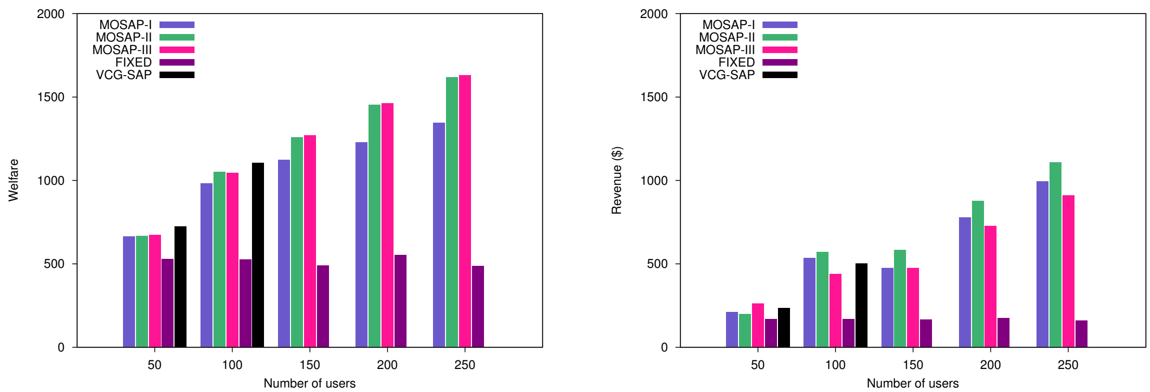


Figure 6: Small-scale experiments with 1 unit capacity: Total allocated units with payment

the first price mechanism, the maximum revenue shown in Fig. 4a cannot be guaranteed. This is due to the fact that the users would not bid their true values and by lowering their bids, the revenue of the provider would decrease. As discussed in Section 3, the expected revenue in our proposed mechanisms and their respective first price versions is the same, while our proposed mechanisms are strategy-proof.

Fig. 5a shows the execution times of the mechanisms on a logarithmic scale. As we expected, the execution time of MOSAP-X and FIXED are very small. This is due to the fact that the time complexity of MOSAP-X and FIXED is polynomial in the size of input. The results show that MOSAP-X is suitable for providing charging services in realtime. Note that small execution time of online charging mechanisms is a must have property in such settings. However, the execution time of VCG-SAP, is more than five orders of magnitude greater than that of MOSAP-X.



(a) (b)

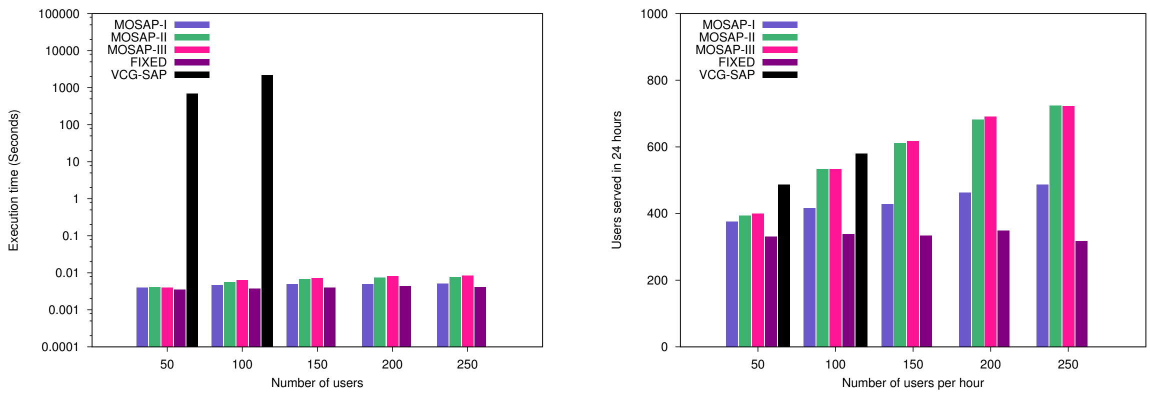
Figure 7: Large-scale experiments: (a) Welfare; (b) Revenue. (*VCG-SAP was not able to determine the allocation for the number of users higher than 100 in feasible time, and thus, there are no bars in the plots for the remaining cases)

Fig. 5b shows the average number of served users for the mechanisms. These users are the ones who have their requested charging units fully scheduled. MOSAP-II, MOSAP-III, and VCG-SAP serve more users than MOSAP-I and FIXED. This is due to the fact that the solution determined by MOSAP-II and MOSAP-III are closer to the optimal solution (as it is shown in Fig. 4a). Note that the requested amount of charging by a user can be more than 1 unit.

Fig. 6 shows total allocated units with payment obtained by the mechanisms. The results show that VCG-SAP allocates almost all the available units during the 24 hours to users who receive their entire requests. MOSAP-X is also capable of allocating the entire requests of users close to that of optimal solution. The remaining units are allocated to some users who do not receive their entire requests due to preemption. However, the results obtained by FIXED are far from that of optimal despite the fact that all the units are allocated to users while these users are not necessarily receiving their entire requests.

6.2.2 Large-scale experiments

We analyze the performance of MOSAP-X, VCG-SAP, and FIXED, where the available capacity is 50 units of 10 kWh. In this case, the number of users that arrive every hour is between 50 and 250. For the instance of the problem with more than 100 users in every hour, VCG-SAP was not able to find the optimal solution even after one hour, which is the entire time interval. This is due to fact that the execution time of VCG-SAP becomes prohibitive for large instances of the problem.



(a) (b)

Figure 8: Large-scale experiments: (a) Execution time; (b) Total served users. (*see Fig.7 note on VCG-SAP)

Note that in this online setting, the mechanisms are expected to respond in realtime. As a result, we did not capture the solutions obtained after one hour of execution of the mechanisms.

Fig. 7a shows the welfare obtained by the mechanisms. The results show that MOSAP-II and MOSAP-III obtain a welfare very close to the optimal (obtained by VCG-SAP) in cases with 50 and 100 users. For the remaining cases, MOSAP-II and MOSAP-III obtain the highest welfare among all the mechanisms. Similar to the welfare obtained by the mechanisms in the small-scale experiments presented in Fig 4a, MOSAP-II and MOSAP-III obtain higher welfare than those obtained by MOSAP-I and FIXED mechanisms. As in the case of the small-scale experiments, the welfare obtained by MOSAP-I is not close to the optimal results because it does not consider the amount of requested charging units by users in its scheduling function. FIXED also obtains welfare far from the other mechanisms. Fig. 7b shows the revenue obtained by the provider when using the mechanisms. Note that the VCG-SAP is optimal in terms of welfare and not the revenue. The results show that MOSAP-II obtains the highest revenue among all the mechanisms in most cases. These results are in agreement with the results presented in Fig 4b.

Fig. 8a shows the execution times of the mechanisms on a logarithmic scale. The execution time of MOSAP-X and FIXED are very small, in the order of milliseconds. However, the execution time of VCG-SAP, is more than five orders of magnitude greater than that of MOSAP-X in the first two cases. A comparison of the execution time of VCG-SAP between small-scale and large-scale experiments shows that the execution time of VCG-SAP grows exponentially when the available

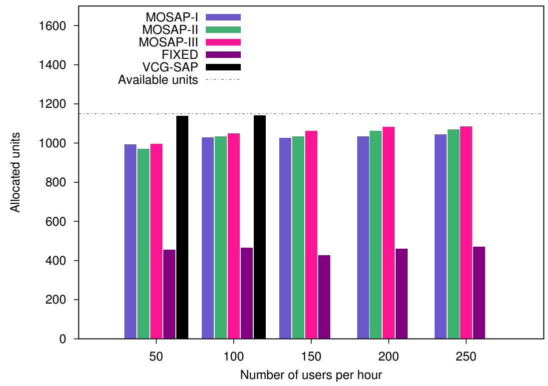
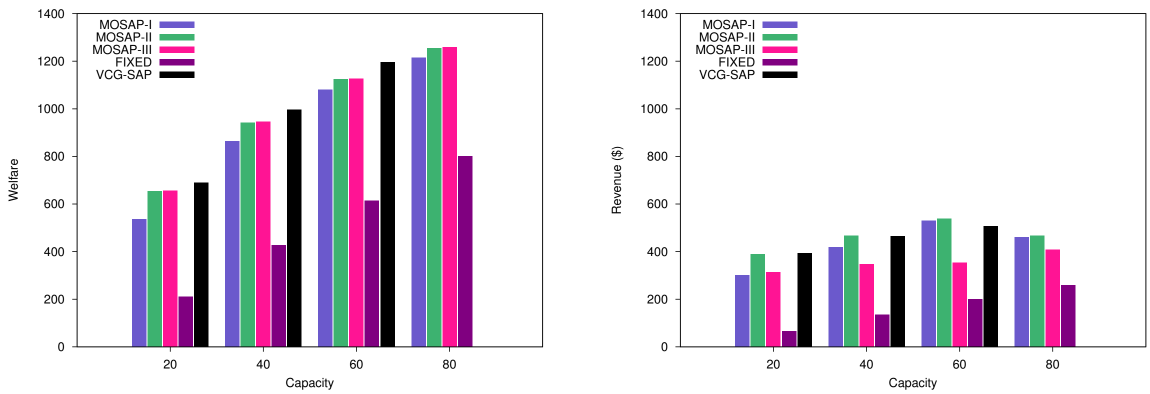


Figure 9: Large-scale experiments: Total allocated units with payment (*see Fig.7 note on VCG-SAP)

charging capacity and the number of users increase. The fact that the execution time of MOSAP-X even for large-scale experiments is in terms of milliseconds make it suitable to be incorporated in online charging settings. Fig. 8b shows the average number of served users whose entire requests are scheduled by the mechanisms. MOSAP-II, MOSAP-III, and VCG-SAP serve more users than MOSAP-I and FIXED. MOSAP-I selects the users only based on their values with no consideration for the requested amount of charging units. This prevents MOSAP-I to serve a higher number of users than MOSAP-II and MOSAP-III, given the limited amount of charging capacity.

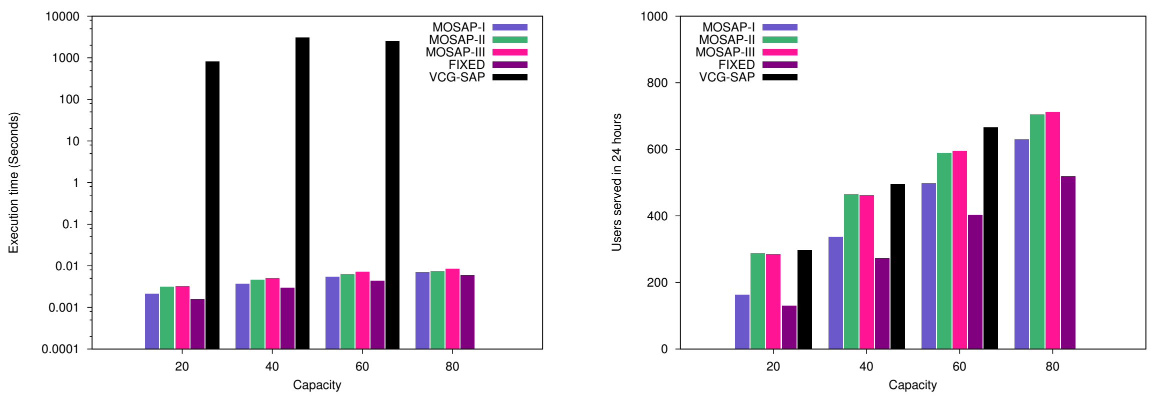
Fig. 9 shows total allocated units with payment obtained by the mechanisms. These results show that VCG-SAP allocates almost all the available units during the 24 hours to users who receive their entire requests. MOSAP-X is also capable of allocating the entire requests of users close to that of optimal solution. The remaining units are allocated to some users who do not receive their entire requests due to preemption. However, the results obtained by FIXED are far from the optimal despite the fact that all the units are allocated to users while these users are not necessarily receiving their entire requests.

From the results of these experiments we can conclude that MOSAP-II obtains on average higher revenue than the other mechanisms, while at the same time finds solutions close to the optimal solutions obtained by VCG-SAP. MOSAP-X finds the charging schedule and payment of users much faster than VCG-SAP. From the results of these experiments we can conclude that MOSAP-X is very suitable for utility providers, since it allows them to make decisions in real-time.



(a) (b)

Figure 10: Sensitivity analysis of available capacity: (a) Welfare; (b) Revenue.



(a) (b)

Figure 11: Sensitivity analysis of available capacity: (a) Execution time; (b) Total served users.

6.2.3 Sensitivity analysis on capacity

To show the effects of change in capacity on the performance of MOSAP-X, we perform sensitivity analysis with respect to capacity. For this set of experiments, the number of users that arrive every hour is 100, while the capacity per hour is varied between 20 and 80 units. In this setting, VCG-SAP could not find the optimal solution in one hour when the capacity is 80. As a result, there is no bar for VCG-SAP in Fig. 10-Fig. 12 for the case of 80 units.

Fig. 10a shows the welfare obtained by the mechanisms. The results show that MOSAP-II and MOSAP-III obtain a welfare very close to optimal (obtained by VCG-SAP). By increasing the capacity, the obtained welfare by all the mechanisms increases since more users can be served.

Fig. 10b shows the revenue obtained by the provider, where MOSAP-II obtains the highest

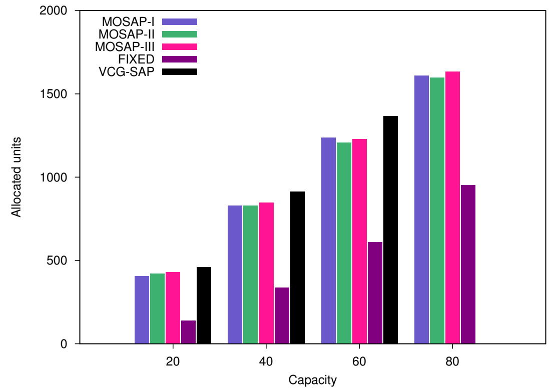


Figure 12: Sensitivity analysis of available capacity: Total allocated units with payment

revenue among all the mechanisms. By increasing the capacity from 20 to 60, the revenue obtained by the provider increases for all the mechanisms. However, when the capacity is 80 units, we do not observe such increase in the revenue. This is due to the fact that when the supply is high, and the mechanisms may be able to fulfill more requests, the price of charging units can decrease leading to a lower revenue.

Fig. 11a shows the execution times of the mechanisms on a logarithmic scale. The execution time of MOSAP-X and FIXED are very small. The execution time of VCG-SAP does not necessarily increase with the increase in capacity since finding optimal solutions for the problem instances with lower capacity may need more time. Fig. 11b shows the average number of served users whose entire requests are scheduled by the mechanisms. The results show that the number of served users increases by all the mechanisms with the increase in capacity. Fig. 12 shows total allocated units with payment obtained by the mechanisms. These results show that MOSAP-X is capable of allocating the entire requests of users close to that of optimal solution.

In real world settings, both the capacity of the utility provider and the arrival rate of charging requests can vary over time. We design our experiments to analyze both of these scenarios. In the small-scale and large-scale experiments, the number of requests changes while we choose a fixed amount of capacity. We also perform a sensitivity analysis on capacity while the request arrival rate is fixed. From all experiments, we conclude that MOSAP-X is capable of providing online scheduling and pricing services in real world settings. These results show that, MOSAP-X also provides these services obtaining high revenue, close to optimal welfare, small execution time, while at the same time, users do not need to strategize to interact with the mechanism.

7 Conclusion

The dynamics of charging requests and the fact that utility providers need to consider load balancing necessitates designing preemption-aware online mechanisms for EV charging. In this paper, we proposed a framework for EV charging considering the incentives of both utility providers and EV drivers. Our proposed framework brings about a win-win situation in which EV drivers can receive their charging requests at lower prices, and utility providers can sell their unused capacity while considering their load balancing objectives. We introduced the problem of online scheduling and pricing for EV charging, and designed a family of online mechanisms, MOSAP-X. We proved that our proposed mechanisms are strategy-proof, where truthful reporting is a dominant strategy for users. We performed extensive experiments that showed that the proposed mechanisms are not only capable of finding close to optimal solutions, but are also very fast and obtain high revenue. The promising results make MOSAP-X suitable for scheduling and pricing EV charging in real-time. For future work, we plan to design and investigate new online mechanisms in presence of multiple utility providers.

References

Almuhtady, A., S. Lee, E. Romeijn, M. Wynblatt, and J. Ni (2014). A degradation-informed battery-swapping policy for fleets of electric or hybrid-electric vehicles. *Transportation Science 48*(4), 609–618.

Balcan, M.-F., A. Blum, and Y. Mansour (2008). Item pricing for revenue maximization. In *Proc. of the 9th ACM Conference on Electronic Commerce*, pp. 50–59. ACM.

Bashash, S., S. J. Moura, J. C. Forman, and H. K. Fathy (2011). Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity. *Journal of Power Sources 196*(1), 541–549.

Cao, J., N. Schofield, and A. Emadi (2008). Battery balancing methods: A comprehensive review. In *Proc. of the IEEE Vehicle Power and Propulsion Conference*, pp. 1–6.

Celis, L. E., G. Lewis, M. Mobius, and H. Nazerzadeh (2014). Buy-it-now or take-a-chance: Price discrimination through randomized auctions. *Management Science 60*(12), 2927–2948.

Chawla, S., J. D. Hartline, D. Malec, and B. Sivan (2013). Prior-independent mechanisms for scheduling. In *Proc. 45th Annual ACM Symp. on Theory of Computing*, pp. 51–60.

Chocteau, V., D. Drake, P. R. Kleindorfer, R. J. Orsato, and A. Roset (2011). Sustainable fleet operations: The collaborative adoption of electric vehicles. *INSEAD Faculty and Research Working Paper*.

Clarke, E. (1971). Multipart pricing of public goods. *Public Choice 11*(1), 17–33.

Clement, K., E. Haesen, and J. Driesen (2009). Coordinated charging of multiple plug-in hybrid electric vehicles in residential distribution grids. In *Proc. of the IEEE/PES Power Systems Conference and Exposition*, pp. 1–7.

Conitzer, V. and T. Sandholm (2004). Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proc. of the 5th ACM Conference on Electronic Commerce*, pp. 132–141. ACM.

Dughmi, S. and T. Roughgarden (2014). Black-box randomized reductions in algorithmic mechanism design. *SIAM Journal on Computing 43*(1), 312–336.

Fairley, P. (2010). Speed bumps ahead for electric-vehicle charging. *IEEE Spectrum 47*(1), 13–14.

Flath, C. M., J. P. Ilg, S. Gottwalt, H. Schmeck, and C. Weinhardt (2014). Improving electric vehicle charging coordination through area pricing. *Transportation Science 48*(4), 619–634.

Gan, L., U. Topcu, and S. Low (2013). Optimal decentralized protocol for electric vehicle charging. *IEEE Transactions on Power Systems 28*(2), 940–951.

Gerding, E. H., V. Robu, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings (2011). Online mechanism design for electric vehicle charging. In *Proc. of the 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 811–818.

Gerding, E. H., S. Stein, V. Robu, D. Zhao, and N. R. Jennings (2013). Two-sided online markets for electric vehicle charging. In *Proc. of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 989–996.

Gershkov, A. and B. Moldovanu (2010). Efficient sequential assignment with incomplete information. *Games and Economic Behavior 68*(1), 144–154.

Glerum, A., L. Stankovikj, M. Thémans, and M. Bierlaire (2014). Forecasting the demand for electric vehicles: accounting for attitudes and perceptions. *Transportation Science 48*(4), 483–499.

Groves, T. (1973). Incentives in teams. *Econometrica: Journal of the Econometric Society 41*(4), 617–631.

Hajiaghayi, M., R. Kleinberg, M. Mahdian, and D. C. Parkes (2005). Online auctions with re-usable goods. In *Proc. of 6th ACM Conference on Electronic Commerce*, pp. 165–174. ACM.

Hartline, J. and A. Karlin (2007). Profit maximization in mechanism design. In N. Nisan, T. Roughgarden, Éva Tardos, and V. V. Vazirani (Eds.), *Algorithmic Game Theory*. Cambridge, UK: Cambridge Univ. Press.

Heydenreich, B., R. Müller, and M. Uetz (2007). Games and mechanism design in machine scheduling: an introduction. *Production and Operations Management 16*(4), 437–454.

Huang, J., M. Leng, L. Liang, and J. Liu (2013). Promoting electric automobiles: supply chain analysis under a governments subsidy incentive scheme. *IIE Transactions 45*(8), 826–844.

Hybrid Vehicle Timeline (2012). `http://www.hybridcenter.org/hybrid-timeline.html`.

Jin, C., J. Tang, and P. Ghosh (2013). Optimizing electric vehicle charging: a customer's perspective. *IEEE Transactions on Vehicular Technology 62*(7), 2919–2927.

Kasbekar, G. S. and S. Sarkar (2010). Spectrum auction framework for access allocation in cognitive radio networks. *IEEE/ACM Transactions on Networking 18*(6), 1841–1854.

Kieckhäfer, K., T. Volling, and T. S. Spengler (2014). A hybrid simulation approach for estimating the market share evolution of electric vehicles. *Transportation Science 48*(4), 651–670.

Lavi, R. and N. Nisan (2004). Competitive analysis of incentive compatible on-line auctions. *Theoretical Computer Science 310*(1-3), 159–180.

Lin, Z. (2014). Optimizing and diversifying electric vehicle driving range for us drivers. *Transportation Science 48*(4), 635–650.

Mak, H.-Y., Y. Rong, and Z.-J. M. Shen (2013). Infrastructure planning for electric vehicles with battery swapping. *Management Science 59*(7), 1557–1575.

Mashayekhy, L., M. Nejad, and D. Grosu (2015a). Physical machine resource management in clouds: A mechanism design approach. *IEEE Transactions on Cloud Computing 3*(3), 247–260.

Mashayekhy, L., M. Nejad, and D. Grosu (2015b). A PTAS mechanism for provisioning and allocation of heterogeneous cloud resources. *IEEE Transactions on Parallel and Distributed Systems 26*(9), 2386–2399.

Misra, S., S. Bera, and T. Ojha (2015). D2p: Distributed dynamic pricing policy in smart grid for phevs management. *IEEE Transactions on Parallel and Distributed Systems 26*(3), 702–712.

Mu'Alem, A. and N. Nisan (2008). Truthful approximation mechanisms for restricted combinatorial auctions. *Games and Economic Behavior 64*(2), 612–631.

Myerson, R. B. (1981). Optimal auction design. *Mathematics of Operations Research 6*(1), 58–73.

Nejad, M., L. Mashayekhy, and D. Grosu (2015). Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds. *IEEE Transactions on Parallel and Distributed Systems 26*(2), 594 – 603.

Nejad, M., L. Mashayekhy, D. Grosu, and R. Chinnam (2016). Optimal routing for plug-in hybrid electric vehicles. *Transportation Science (accepted)*.

Nisan, N. (2007). Introduction to mechanism design. In N. Nisan, T. Roughgarden, Éva Tardos, and V. V. Vazirani (Eds.), *Algorithmic Game Theory*. Cambridge, UK: Cambridge Univ. Press.

Parkes, D. C. (2007). Online mechanisms. In N. Nisan, T. Roughgarden, Éva Tardos, and V. V. Vazirani (Eds.), *Algorithmic Game Theory*. Cambridge University Press.

Parkes, D. C. and S. Singh (2003). An MDP-based approach to online mechanism design. In *Proc. of 17th Annual Conference on Neural Information Processing Systems*.

Porter, R. (2004). Mechanism design for online real-time scheduling. In *Proc. of 5th ACM Conference on Electronic Commerce*, pp. 61–70. ACM.

Robu, V., E. H. Gerding, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings (2013). 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.

Robu, V., S. Stein, E. H. Gerding, D. C. Parkes, A. Rogers, and N. R. Jennings (2012). An online mechanism for multi-speed electric vehicle charging. In *Auctions, Market Mechanisms, and Their Applications*, pp. 100–112. Springer.

Sandholm, T. and A. Likhodedov (2015). Automated design of revenue-maximizing combinatorial auctions. *Operations Research 63*(5), 1000–1025.

Schneider, M., A. Stenger, and D. Goeke (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science 48*(4), 500–520.

Sioshansi, R. (2012). Or forum-modeling the impacts of electricity tariffs on plug-in hybrid electric vehicle charging, costs, and emissions. *Operations Research 60*(3), 506–516.

Stein, S., E. Gerding, V. Robu, and N. R. Jennings (2012). A model-based online mechanism with pre-commitment and its application to electric vehicle charging. In *Proc. of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 669–676. International Foundation for Autonomous Agents and Multiagent Systems.

Stüdli, S., W. Griggs, E. Crisostomi, and R. Shorten (2014). On optimality criteria for reverse charging of electric vehicles. *IEEE Transactions on Intelligent Transportation Systems 15*(1).

Sundstrom, O. and C. Binding (2012). Flexible charging optimization for electric vehicles considering distribution grid constraints. *IEEE Transactions on Smart Grid 3*(1), 26–37.

Teslamotors (2015). `https://www.teslamotors.com/models-charging`.

Vasirani, M. and S. Ossowski (2012). A computational monetary market for plug-in electric vehicle charging. In *Auctions, Market Mechanisms, and Their Applications*, pp. 88–99. Springer.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance 16*(1), 8–37.

Wei, L. and Y. Guan (2014). Optimal control of plug-in hybrid electric vehicles with market impact and risk attitude. *Transportation Science 48*(4), 467–482.

Yilmaz, M. and P. T. Krein (2013). Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles. *IEEE Transactions on Power Electronics 28*(5), 2151–2169.

Zhou, X., S. Gandhi, S. Suri, and H. Zheng (2008). ebay in the sky: strategy-proof wireless spectrum auctions. In *Proc. of the 14th ACM International Conference on Mobile Computing and Networking*, pp. 2–13.

Biographies

Mark M. Nejad is an assistant professor in the School of Industrial and Systems Engineering at the University of Oklahoma. He received his Ph.D. in industrial engineering and M.Sc. in computer science from Wayne State University. His research interests include network optimization, game theory, electric vehicles, distributed data analytics, and cloud computing. He received the 2016 Institute of Industrial and Systems Engineers (IISE) Pritzker Best Doctoral Dissertation Award. He also received several best paper awards including: the 2014 INFORMS ENRE best student paper award, and runner-up awards for the 2014 POMS College of Sustainable Operations Best Student Paper Competition and the 2014 INFORMS Service Science Best Paper Award. He is a member of the INFORMS, IISE, and IEEE.

Lena Mashayekhy is an assistant professor in the Department of Computer and Information Sciences at the University of Delaware. She received her PhD degree in computer science from Wayne State University in 2015. Her research interests include big data and cloud computing, distributed systems and parallel algorithms, algorithmic game theory, and electric vehicles. She has published more than thirty peer-reviewed papers in venues such as IEEE Transactions on Computers and Transportation Science. She is a member of the IEEE, ACM, and INFORMS.

Ratna Babu Chinnam is a Professor in the Department of Industrial & Systems Engineering at Wayne State University (Detroit, U.S.A.). He is the author of over 150 journal articles, conference proceedings, and technical reports in the areas of Supply Chain Management, Big Data Analytics, Sustainability, and Diagnostics/Prognostics. Most of his research is funded by the U.S. federal agencies (NSF, DOT, DOD, DOE, VA) and he carried out extensive funded collaborative research with international companies such as Ford Motor Company, General Dynamics, Chrysler and consulted for such companies as Dominos, Urban Science, Energy Conversion Devices, Federal Mogul, and Sirius Satellite Radio. He supervised over 24 PhD students that mostly occupy academic positions around the world. He is a member of Alpha Pi Mu, INFORMS, and NAMRI.

Daniel Grosu received the Diploma in engineering (automatic control and industrial informatics) from the Technical University of Iaşi, Romania, in 1994 and the MSc and PhD degrees in computer science from the University of Texas at San Antonio in 2002 and 2003, respectively. Currently, he is an associate professor in the Department of Computer Science, Wayne State University, Detroit. His research interests include parallel and distributed systems, cloud computing, parallel algorithms, resource allocation, computer security, and topics at the border of computer science, game theory and economics. He has published more than ninety peer-reviewed papers in the above areas. He has served on the program and steering committees of several international meetings in parallel and distributed computing. He is a senior member of the ACM, the IEEE, and the IEEE Computer Society.