Large Scale Charging of Electric Vehicles

Shiyao Chen, Yuting Ji, and Lang Tong
School of Electrical and Computer Engineering
Cornell University, Ithaca, NY 14850
{sc933,yj246,lt35}@cornell.edu

Abstract—**The problem of scheduling for large scale charging of Electric Vehicles (EVs) is considered. As part of the future EV infrastructure, a Large Scale Charging (LSC) facility is capable of charging hundreds of electric vehicles simultaneously. As an intelligent load in the future smart grid, LSC requires properly designed pricing and scheduling algorithms that take into account the electricity consumed, the arrival-departure characteristics, and overall charging capacity.**

The scheduling of LSC is formulated as a deadline scheduling problem. Utility functions that combine both amount of charge and tightness of the deadline are proposed. Under arbitrary (and deterministic) arrival, departure, and charging characteristics, a scheduling policy referred to as deadline scheduling with admission control is proposed. The proposed algorithm achieves the highest competitive ratio (against the best offline scheduling) for the utility function linear in charging level among all online scheduling algorithms. It also offers significant gain over benchmark scheduling algorithms such as the Earliest Deadline First (EDF) scheduling and the First Come First Serve (FCFS) scheduling in terms of average performance for general utility functions when tested with randomly generated charging requests.

Index Terms—**Electric vehicle infrastructure, large scale charging, EV/PHEV charging, demand response.**

I. Introduction

The electrification of the transportation system is one of the key components toward a clean and sustainable society. The technology for Electric Vehicles[1] (EVs) has sufficiently advanced that an accelerated adoption of EVs is increasingly likely. Crucial to the transition toward an EV based transportation is to establish Large Scale Charging (LSC) infrastructures, *i.e.*, a battery charging system at public parking facilities, work places, and apartment complexes where a large number of EVs are charged simultaneously. LSC is essential for urban areas, especially in densely populated developing countries, where in-home EV charging is not an option. With intelligent scheduling, LSC sites can participate in demand side response, taking advantages of the economies of scale, and offering greater energy efficiency. For the management of the grid, LSC provides an opportunity for efficient load management [1], [2].

The pricing and scheduling of large scale EV charging facility is considered in this paper. Specifically, a pricing scheme coupled with a deadline scheduling algorithm that exploits the available charging capacity and the customer's flexible schedule is proposed. It is assumed that a customer arriving at a charging facility can communicate its charging needs (the amount of charging required and the deadline for completion) to the operator. The pricing of a charging request should be a function of not only the amount of required charging but also the deadline of completion. To an LSC operator, a job with a relaxed deadline offers the scheduling flexibility to accommodate more profitable requests. Such requests also give the operator the opportunities to avoid power surge and exploit future pricing advantages. Therefore, customers should be given price incentives to offer their flexibilities.

The scheduling of LSC is studied under the framework of online scheduling of jobs with deadlines. In the online scheduling setting there is no reservation requirements, and the operator has to decide, at the time of customer arrival with a charging request, whether to accept the customer and the price of service. In a more general setting, the pricing itself can be used as a way to reject a request or reduce the amount of requests. In this paper, however, the policy of admission is considered separately from that of pricing.

For online deadline scheduling algorithms operating in uncertain environments, it is often inevitable that some jobs cannot be completed by their deadlines. In the context of LSC, for example, the power shortage or faulty equipments may affect the completion of accepted charging requests. The admission control can be used as a way to reduce but not eliminate such occurrences. Therefore, there is a need, as part of the pricing scheme, for a way to compensate the customer whose charging request is accepted but not completed or not to the requested level. For instance, a voucher for free future services or cash credits for the uncharged amount can be offered. Such mechanisms of compensating the customers with unfulfilled requests need to be incorporated to the design of optimal scheduling algorithm.

A. Summary of Results

The scheme presented in this paper focuses on the pricing and the online scheduling for LSC, which is the first of this kind to our best knowledge. An *arbitrary (non stochastic)* model for arrival, departure, and charging processes is adopted. Such a non-stochastic setting is appealing for both mathematical and practical reasons. Because LSC for EVs is still at an embryonic state, credible models that characterize the processes of charging requests, the duration of stay, and the

This work is supported in part by the National Science Foundation under Grant CNS-1135844.

[1] Include both Plug-in Hybrid Electric Vehicles (PHEVs) and Battery Electric Vehicles (BEVs).

level of charging are missing. Since the setting used in this paper is assumed arbitrary, the proposed solution does not require the knowledge of specific parameters.

A utility function and an associated pricing scheme that ties both the amount of charging and the urgency of the deadline are proposed. In particular, for a customer arriving at time r with a charging request of p (joules) and deadline d, the utility of such a job to the operator is $u(p, \frac{d-r-p}{p})$ where u is increasing with respect to the amount of charging p for fixed $\frac{d-r-p}{p}$ (the relative deadline factor) and decreasing with respect to $\frac{d-r-p}{p}$ for fixed p. Following the proposed pricing scheme the customer will be charged for the amount of the utility u if the request is fulfilled by the deadline. If the operator fails to fulfill the request, the operator compensates the customer in an amount proportional to the unfulfilled amount (negative price for the customer).

An online scheduling algorithm aiming at maximizing the reward under arbitrary arrival, departure (deadline), and charging instances is proposed. Referred to as Deadline Scheduling with Admission Control (DSAC), this algorithm extends an earlier algorithm proposed in [3] for problems involving multiple chargers. This generalization is necessary when parallel charging—an essential feature for LSC—is allowed.

To measure performance of algorithms under arbitrary deterministic formulations, it is standard to use *competitive ratio* analysis, which compares an online algorithm against the best possible offline scheduling algorithm among all *charging instances*. It can be shown that DSAC is optimal for linear utility functions ($u \propto p$, or $u = kp$) in terms of competitive ratio. In other words, DSAC scheduler provides the best performance guarantee under the worst possible charging scenarios. It is also demonstrated that DSAC offers significant improvements in *average performance* over benchmark algorithms such as the Earliest Deadline First (EDF) and the First Come First Serve (FCFS) algorithms for general utility functions.

B. Related Work

The benefits and impact of EV on the electricity network are by now well documented. See, *e.g.,* [4], [5], [6] where the authors concluded that the current generation capacity is able to sustain the additional EV charging. System level constraints for the electric grid after introducing EV charging are considered in [7], [8], [9]. There has also been economic analysis of EV charging technology. The authors of [10] has conducted an energy economic analysis of EV charging using solar photovoltaic panels at workplace parking garage with the conclusion that EV charging facility in public garage is economically beneficial to both the car owners as well as the facility operator.

For the scheduling of EVs, a variety of modeling and optimization techniques have been proposed in the literature. The EV charging for public garages has been considered in [11], [12]. Specifically, the authors of [11] aggregated system and operation models for the simulation of EV charging in a municipal parking lot. The method of particle swarm optimization is employed to allocate energy to EVs in [12]. The techniques in [11], [12] are significantly different from the approach proposed in this paper, and there is no stated performance guarantee; performance can only be evaluated by simulations.

The proposed algorithm DSAC is rooted in the classical deadline scheduling problem, stemming from the earlier problems of scheduling of processors for computer systems. It should be noted, however, that the standard deadline scheduling formulation does not include penalty if the job is not completed by the scheduler. Since the seminal work of Liu and Layland [13], there is considerable literature addressing the scheduling problem in the underloaded and overloaded regimes. The former corresponds to the case when there exists an offline scheduling algorithm that can complete all jobs arrived whereas the latter corresponds to the case when some jobs cannot be completed even for the best offline scheduling algorithm. For the underloaded scenario with single processor, it has been shown that simple online scheduling algorithms such as Earliest Deadline First (EDF) [13], [14] and Least Laxity First (LLF) [15] achieve the same performance as the optimal offline scheduling algorithm. The assumption of underloaded overall workload, however, is restrictive and unverifiable in practice. Locke showed in [16] that both EDF and LLF can perform poorly in the presence of overload. There were efforts to develop an online scheduling algorithm with worst case performance guarantee when the system is overloaded in [17], [18].

The admission control has been considered in the literature under the term “immediate notification” in [19] for the application of video-on-demand. Later, joint admission and scheduling has been studied in [20], [21], [22], [23] for non-preemptive job requests without non-completion penalty. The authors of [24], [25] give separately two joint admission and scheduling algorithms for preemptive job requests with non-completion penalty under proportional value model ($v \propto p$), where the non-completion penalty is the entire request value v.

The deadline scheduling algorithm proposed in this paper is an extension of an approach in [3] where its application to large scale charging was mentioned but not addressed explicitly. There are major differences between [3] and the current paper in both the application setting and the specific techniques including the utility and reward structure, the presence of multiple processors and parallel charging, and the evaluation of average performance in performance comparison.

II. EV Charging Model

A. Charging Infrastructure and Customer Requests

The charging infrastructure at the facility consists of m charging plugs (processors) with constant charging speed. In LSC operations, preemption is allowed at no cost, *i.e.*, a preempted battery can be resumed charging from the previous battery level upon preemption.

Each EV charging request $T = (r, p, d)$ is represented by a triple specified by the arrival (release) time r, charging (processing) time p and deadline d. For example, a customer who lives in an apartment in a high-rise building without overnight charging equipment may arrive at a EV charging facility near his office building around 8 am on the way to work. The customer may intend to catch a flight for a conference at 2 pm and plan to leave for the airport at 12 pm. The current battery level may be 10 miles and in order to make the round trip to the airport the desired battery level after charging is 50 miles. In this example the release time is 8 am, the deadline is 12 pm and the processing time is determined by the 40 miles desired battery level as well as the charging speed of the charging plug.

Over a certain period of time, *e.g.*, one day, all the customer requests submitted constitute the input sequence $I = (T_1, T_2, \ldots, T_n)$ including charging requests $T_1 = (r_1, p_1, d_1), \ldots, T_n = (r_n, p_n, d_n)$ to be priced, admitted and scheduled by the LSC facility operator. Each fulfilled customer request is associated with an individual utility, which is a given function $u(r, d, p)$ of the request parameters r, d and p, whereas each admitted but unfulfilled request is associated with a negative individual utility $u(r, d, \hat{p})$, where $\hat{p}$ is the unfulfilled amount of the requested charging level.

Given the charging infrastructure and customer requests model, the problem of interest is to devise a mechanism of LSC operation to maximize the collective utility of the customers. To this end we adopt the approach that uses the revenue-seeking LSC operator as a proxy to maximize the collective customer utility. Specifically, the LSC operator prices the customer requests according to the individual utility and conducts the admission control and scheduling in a revenue-seeking fashion for his own benefit. When the individual utility function is not known or known with uncertainty to the LSC operator, the impact of using a pricing function that deviates from the true individual utility function will be investigated via simulation.

B. Interaction between Customers and LSC Operator

The interaction between the customers and the LSC facility operator is summarized in the price quote offered by the LSC operator. After the facility operator is given the charging request parameters r, p and d, the facility operator offers a price v for the charging request. The objective of the LSC operator is to maximize the revenue, whereas the objective of the customer is to obtain battery charging at a reasonable price. In the process of both parties maximizing their own utility, our system model allows the LSC operator to decline a customer request (*e.g.*, because the facility is currently busy serving more profitable requests) to protect the utility of the LSC operator, and thus indirectly expands the collective customer utility by allocating the time and charging infrastructure to the requests with better individual utility. Our system model also allows the customer to evaluate the price quote and decide to seek charging elsewhere. However, once the offered price quote is accepted by the customer a contract is established between the LSC operator and the customer; if the accepted charging request is not completed by its deadline as promised, the LSC operator loses the associated value (quoted price) of the request and has to pay an additional penalty depending on the amount of unfinished charging level. Specifically, the non-completion penalty is equal to $v\frac{\hat{p}}{p}$, where $\hat{p}$ denotes the unfinished charging level, *i.e.*, the non-completion penalty is the fraction in the quoted price that corresponds to the unfinished charging level. This non-completion liability protects the utility of the customer. This specific non-completion liability suits LSC well since utility is delivered to the EV owner continuously as the battery charging level increases.

The profit obtained by the LSC operator is the total value of all completed charging requests before their deadlines, less all penalties paid for the admitted requests that miss their deadlines. The pricing, admission and the scheduling has to be conducted in an online fashion, *i.e.*, the LSC operator knows the parameters of request T_i only at the release time r_i. The LSC operator strives to design an online management scheme with satisfactory performance in both underloaded and overloaded regimes.

III. Pricing Scheme and Utility

The pricing function $v(r, d, p)$ should be tied to the individual utility of the customer request, since this provides an incentive for the customers to consider their flexibility and submit charging requests with relaxed deadlines whenever possible.

The pricing scheme has two effects for the LSC operation.

1) Pricing scheme determines customer response, *i.e.*, shaping the fraction of customers that accept a certain price $v(r, d, p)$ offered for requests with release time r, deadline d and processing time p. Specifically, an unreasonably high price will turn away the majority of the customers and reduce both the revenue of the LSC operator as well as the collective utility of the EV owners. On the other hand, a price too low may overwhelm the charging facility without earning appropriate revenue for the LSC operator.
2) Due to the revenue-seeking nature of the LSC operator, the pricing scheme will affect the specific admission and scheduling decision since different prices may tag different priorities to the charging requests in the view of LSC operator. It is sensible for the LSC operator to devote more resource and time on the customers who accepted more rewarding quoted prices.

The first effect of the pricing scheme leads to the traditional method of pricing a standard product. Specifically, with the knowledge of the customer response curve $f(v; r, d, p)$, where $f(v; r, d, p)$ gives the fraction of customers with release time r, deadline d and processing time p that are willing to accept the price v, the LSC operator maximizes $vf(v; r, d, p)$. This method balances the quoted price with the customer response curve; in both extremes of v the revenue function $vf(v; r, d, p)$ cannot assume the maximum since either v or $f(v; r, d, p)$ is too small.

However, the customer response curve $f(v; r, d, p)$ is difficult to obtain or approximate due to the three additional parameters r, d and p. More importantly, the pricing problem in EV garage charging exhibits significant distinctions from pricing a standard product since, for the standard products manufactured on demand, the contention for manufacturing resource only comes into play in the term of manufacturing cost, even when there are overwhelming orders for the standard product. On the other hand, for LSC application the contention for charging infrastructure and time is explicit. With the limited peak power injection from the electricity network when there are overwhelming requests for charging in a short period of time, the LSC operator simply cannot fulfill all the requests, even at the expense of more operational cost. Therefore, the pricing scheme in EV garage charging serves for the purpose of optimally allocating charging infrastructure and time among charging requests.

Structural properties of the utility function $u = u(r, d, p)$ are presented below.

1) The class of time homogeneous utility functions is considered, *i.e.*, the utility of the request given by the triple (r, d, p) is identical to that of the triple $(r+t, d+t, p)$. It is worth pointing out that there may be additional issues such as peak hour versus non-peak hour. However, in this paper the utility function that adapts to time or the overall customer request arrival process is not considered. One can thus write the utility function $u = u(p, \sigma)$, where $\sigma = (d - r - p)/p$ is the relative deadline factor of the request $T = (r, d, p)$, ranging in $[0, \infty)$, and implies the difficulty for the LSC operator to fulfill the specific charging request without affecting other requests.
2) The utility $u = u(p, \sigma)$ should be an increasing function of the processing time p when σ is fixed, since the electricity consumed is proportional to the charging level requirement p.
3) The utility function $u(r, d, p)$ should be a decreasing function of deadline d when the release time r and the processing length p is fixed, since the extended deadline delays the time the customer can retrieve the car and may reduce the operational difficulty of the LSC operator. This observation translates to the fact that $u = u(p, \sigma)$ is a decreasing function of σ when p is fixed.
4) The decreasing trend of $u = u(p, \sigma)$ in σ can also be interpreted with the interaction among the charging requests that come close in time. A charging request with relative deadline factor very close to 0 cannot afford to be moved around or delayed in the time axis. Therefore stricter commitment in time and charging infrastructure is necessary to fulfill the request which may potentially block or delay other requests. The decreasing trend in the price represents the commitment premium.

Considering the structural properties we adopt the utility function $u = u(p, \sigma) = ph(\sigma)$ in the simulations in Section V, where u is proportional to p, and $h(\sigma)$ is a decreasing function of σ indicating the commitment premium. Exponential function is used for $h(\sigma)$ in Section V.

Ideally the pricing function should be identical to utility function in order to maximize collective utility. However, due to uncertain knowledge of the utility function from LSC operator side we adopt three different types of pricing functions in the form $v = v(p, \sigma) = p\hat{h}(\sigma)$ in the simulations in Section V, where $\hat{h}(\sigma)$ is taken from one of the following three types: constant, exponential (with correct and deviated parameters) and step functions.

IV. Admission and Scheduling

Intuitively, the admission and scheduling tend to be easy for the LSC operator if the overall charging load from the customer requests is well below the facility capacity. Indeed, when the overall charging load is reasonable, simple algorithms such as EDF and first come first serve (FCFS) show reasonable performance by simply admitting all requests that ever arrive. However, if overwhelmingly many charging requests arrive in a short period of time, *e.g.*, during rush hours or due to events like sports games, the admission and scheduling will be more challenging. Described and interpreted below is an online admission and scheduling algorithm DSAC, the performance of which is demonstrated in Section V for both underloaded and overloaded scenarios.

A. DSAC: Admission Control

When a customer request arrives and finds the facility running well below capacity, the LSC operator offers a price quote and once the offered quote is accepted, the operator should immediately admit the customer, dispatch the request to one of the lightly loaded processors and append the request at the end of the current schedule. Otherwise, the charging infrastructure would be left idle and potential profit would be lost. In this easy-to-accommodate situation, the LSC operator essentially takes a greedy approach and notices that admitting the request will bring more revenue for now.

When a customer request arrives to a heavily occupied facility, after the offered price quote is accepted, the LSC operator faces a dilemma that admitting the customer may lead to non-completion liability, while declining the customer again means losing profit at hand. While optimal tradeoff between the two concerns is difficult and involves accommodating the newly arrived customer with minimum non-completion liability, we propose a greedy threshold approach for this difficult-to-accommodate situation.

The key idea behind the admission algorithm for difficult-to-accommodate requests in DSAC is to evaluate the admission decision based on the comparison of the potential profit associated with admitting and declining the customer request. Specifically, the LSC operator enumerates the potential processors. For each processor the admitting option is evaluated by considering the quoted price as well as the incurred potential non-completion penalty; the declining option is evaluated by recognizing the potential value of the requests that would have been affected upon admitting the new request. The ratio of the profit associated with admitting and that with declining is computed for each potential processor. Only if the maximum ratio is over a prescribe threshold, the operator will admit this request and dispatch it to the processor with the maximum ratio.

B. DSAC: Scheduling

Even assuming the LSC operator admits the request just released, there are many alternatives in the specific schedule of the request just released as well as the other pending requests (due to the admission of the new request, it may be necessary to update the schedule of the other requests). DSAC makes the scheduling decision in a greedy manner with minimum backtrack in updating the schedule after admitting the newly released request. Specifically, if the operator decides to admit the request and dispatch to Processor k, the schedule of Processor k is updated by tight-scheduling the newly released request in the interval $[d-p, d]$ where p and d are the processing time and deadline of the newly released request, respectively. Then the part of the previous schedule after time $d-p$ is moved to start at time d, or the end of the current schedule, whichever comes later in time. This moving may lead to some of the moved jobs to miss their deadlines. Therefore the schedule is again updated to remove the part of the moved jobs that comes after their deadlines.

The decision process can be interpreted as follows. When the LSC operator decides to admit the newly released request, the request is profitable once accepted but difficult to accommodate into the current schedule. Therefore in order to accommodate the newly released profitable request, the operator sacrifices the jobs in the current schedule in the time interval $[d-p, d]$, some of which may have deadlines far into the future, thus still have potential in completion even after the moving.

Described next is the procedure to determine the profit associated with admitting (*e.g.*, to Processor 1) and declining the requests that cannot be appended on Processor 1. First execute (virtually) on the current tentative schedule of Processor 1 the procedure associated with the decision to admit the difficult-to-accommodate request (including scheduling the newly released request in $[d-p, d]$ and postponing the previous requests in $[d-p, d]$) and find out the requests in the current tentative schedule of Processor 1 that are affected in the received processing time. Denote by $\mathcal{J}_{\text{affect}}$ the set of requests in the current tentative schedule that are affected in the received processing time. The profit associated with the option of declining can be computed as the value of the subset of requests in $\mathcal{J}_{\text{affect}}$ anticipated to complete by the current tentative schedule, less the portion of penalty attributed to the subset of requests. The profit associated with the option of admitting can be computed as the value of the newly released request, less the portion of penalty attributed to the acceptance of the newly released request (due to affecting the requests in $\mathcal{J}_{\text{affect}}$).

To summarize, the dynamics of DSAC admission and scheduling algorithm can be described as follows: the operator maintains a tentative schedule for each processor at all times; when a customer request is released, the operator checks whether it is possible to append the new request at the end of the current tentative schedule of one of the processors while meeting its deadline. If the deadline can be met, then the request is admitted and appended at the end of the current tentative schedule of that processor. Otherwise, the operator determines whether to admit the request based on the profits of the options of accepting and declining. If the profit associated with accepting is not sufficiently large, then the request is simply declined service. Otherwise, the request is scheduled on the processor with the maximum profit ratio in the time interval $[d_i - p_i, d_i]$; the previous schedule after time $d_i - p_i$ is then moved to start at time d_i, or the end of the current schedule, whichever comes later in time, and the operator further checks whether there are any moved requests that already missed their deadlines after the moving, deletes them and moves the requests accordingly to fill the gap left by the requests deleted.

The pseudo code of DSAC is given below. At time 0 the operator starts the infinite loop in which the schedule is updated upon each request release.

DSAC: Admission and Scheduling procedure

```
 1: loop
 2:     upon event: request $T_{arr}$ is released
 3:     if $T_{arr}$ appendable to Processor $i$ then
 4:         append $T_{arr}$ to the end of the tentative schedule of Processor $i$
 5:     else
 6:         $\hat{\imath} = \arg\max_i \text{Profit}_{i,admit}/\text{Profit}_{i,decline}$
 7:         if $\text{Profit}_{\hat{\imath},admit} > \beta\text{Profit}_{\hat{\imath},decline}$ then
 8:             append $T_{arr}$ at the end by $d_{arr}$ on Processor $\hat{\imath}$
 9:             update the schedule after $d_{arr} - p_{arr}$ accordingly
10:         else
11:             decline $T_{arr}$
12:         end if
13:     end if
14: end loop
```

As indicated in the pseudo code T_{arr} gets admitted and appended to the current schedule on Processor i if it is appendable on Processor i (line 4). Otherwise, if T_{arr} is not appendable to any of the m processors, the profits $\text{Profit}_{i,accept}$ and $\text{Profit}_{i,decline}$ associated with admitting and declining T_{arr} respectively get compared, where i indicates the processor index. For the processor that maximizes the profit ratio (Processor $\hat{\imath}$), if admitting T_{arr} assumes better profit (line 7), then T_{arr} is admitted and appended at the end by d_{arr} (*i.e.*, scheduled in the time interval $[d_{arr} - p_{arr}, d_{arr}]$), and the current schedule after $d_{arr} - p_{arr}$ is moved and modified accordingly (line 8 and 9). Otherwise, if admitting T_{arr} does not have better profit, T_{arr} is declined service (line 11). The threshold β (line 7) represents the tradeoff between the current revenue versus the newly arrived requests.

C. Illustration

Fig. 1 illustrates the admission and scheduling algorithm for the single processor case. The arrows and the circles indicate the release times and the deadlines, respectively. The blue request in Fig. 1(a) and the red request in Fig. 1(e) are appendable and thus admitted and appended in the end of the current schedule. The green request in Fig. 1(b) and the brown request in Fig. 1(f) are difficult to accommodate into the current schedule and thus the profits associated with admitting and declining are examined in Fig. 1(c) and Fig. 1(g). The admission and scheduling decisions in Fig. 1(d) and Fig. 1(h) are rendered by the profit comparison. The schedule is updated by tight-scheduling the green request and the brown request in Fig. 1(d) and Fig. 1(h). The moving caused by the green request and the brown request leads to non-completion penalty of the blue request and the green request, respectively. On the other hand, the red request is still feasible after the delay caused by the brown request in Fig. 1(h).

D. Competitive Ratio Guarantee

In the EV charging problem, an online operator knows the job parameters at the release time, whereas the offline operator is clairvoyant and knows the entire input instance a priori. Due to the prior knowledge of the job parameters, the offline operator is never subject to any non-completion liability. When faced with overloaded charging request sequences, the online operator is at significant disadvantage compared with the offline operator.

To measure the worst case performance guarantee of an online algorithm the metric of competitive ratio is widely used (see Definition 1).

Definition 1. ***Competitive ratio:*** *An online algorithm A is α-competitive for an input set $\mathcal{I}$ if $\min_{I\in\mathcal{I}} \frac{A(I)}{opt(I)} \geq \alpha$ where I varies over all possible input instances in $\mathcal{I}$, and $A(I)$ and $opt(I)$ are the values the online algorithm A and the optimal offline algorithm obtain on input instance I, respectively.*

Specifically, an α-competitive online algorithm is guaranteed to achieve at least α fraction of the optimal offline value under any input instance I in the input set $\mathcal{I}$. In the EV charging problem an input instance I includes a sequence of charging requests $T_1 = (r_1, d_1, p_1, v_1), \ldots, T_n = (r_n, d_n, p_n, v_n)$, and the input instance set $\mathcal{I}$ contains all input instances with finite number of charging requests.

The worst case performance guarantee of the DSAC algorithm in terms of competitive ratio for linear utility function ($u \propto p$, or $u = kp$) is stated in Theorem 1.

Theorem 1. *DSAC algorithm achieves the optimal competitive ratio in the entire set of online algorithms for linear utility function.*

The result in Theorem 1 extends the previous competitive ratio optimality result in [3]. Due to space limit the detailed proof and exposition will be included in future work [26].

V. Average Performance: Simulation Results

A. Simulation Setup

In this section the simulation of the average performance of the DSAC algorithm and comparison with EDF and FCFS with or without admission control are presented. The benefit of appropriate pricing scheme is also demonstrated via simulating the collective utility of the customers.

The system parameters in the simulation are adopted as follows; the customer arrival process is assumed to be Poisson process with parameter λ, the charging level requirements assumed to be i.i.d. uniform random variables in the interval $[5, 100]$, and the relative deadlines assumed to be i.i.d exponential random variables with parameter η, where the relative deadline of charging request $T = (r, d, p)$ is $d - r - p$, *i.e.*, the maximum amount of time that can be spent on other requests in interval $[r, d]$ in order to finish T in time.

The customer utility function is set to be $u(p, \sigma) = ph(\sigma)$, where $h(\sigma)$ is an exponential function ($h(\sigma) = 1 + 3\exp(-\sigma)$). Different pricing functions are simulated with the form $v(p, \sigma) = p\hat{h}(\sigma)$, where $\hat{h}(\sigma)$ is chosen from constant ($\hat{h}(\sigma) = 1$), correct exponential ($\hat{h}(\sigma) = 1 + 3\exp(-\sigma)$), deviated exponential ($\hat{h}(\sigma) = 1 + 2\exp(-\sigma)$), and step functions ($\hat{h}(\sigma) = 1 + 3 \cdot 1_{\{\sigma \leq 2\}}$). In the simulation 200 Monte Carlo runs are conducted with 1000 jobs released over time for each Monte Carlo run. The tradeoff parameter of DSAC algorithm is taken to be $\beta = 2.4$. The EDF and FCFS algorithms are adapted to the multi-processor situation in that an admitted request is dispatched to the processor with shortest current tentative schedule.

The admission control scheme of EDF and FCFS is implemented as follows. Under EDF and FCFS with admission control, the LSC operator declines a newly released request if overload is detected once the request is incorporated into the current schedule. That is, the facility operator checks the feasibility of the requests accepted so far to ensure no penalty will be incurred. If accepting a newly released request may lead to penalty, then the facility operator decides to decline the request. In Fig. 2 the performance of EDF and FCFS is normalized by the value obtained by DSAC.

B. Admission Control: Underloaded and Overloaded Scenarios

Both underloaded and overloaded scenarios are simulated and EDF and FCFS without admission control are compared with DSAC. It can be observed that in lightly loaded scenarios (Fig. 2(a) and 2(c)) EDF and FCFS without admission control perform reasonably well compared with DSAC, especially in the 6-processor situation in Fig. 2(c), where the system load is extremely light and EDF and FCFS without admission control achieves almost 100 percent of the value obtained by DSAC.

On the other hand, in overloaded scenarios (Fig. 2(b) and 2(d)) EDF and FCFS without admission control perform poorly compared with DSAC. When the system load is extremely heavy due to the fact that no admission control is performed, EDF and FCFS admit all the requests and incur excessive non-completion penalty. Therefore the value obtained turns out to be negative, whereas DSAC still earns positive revenue. The pricing function is set to be the true utility function.

The effect of the admission ingredient is thus illustrated by the significant gap in the overloaded scenario. Also the

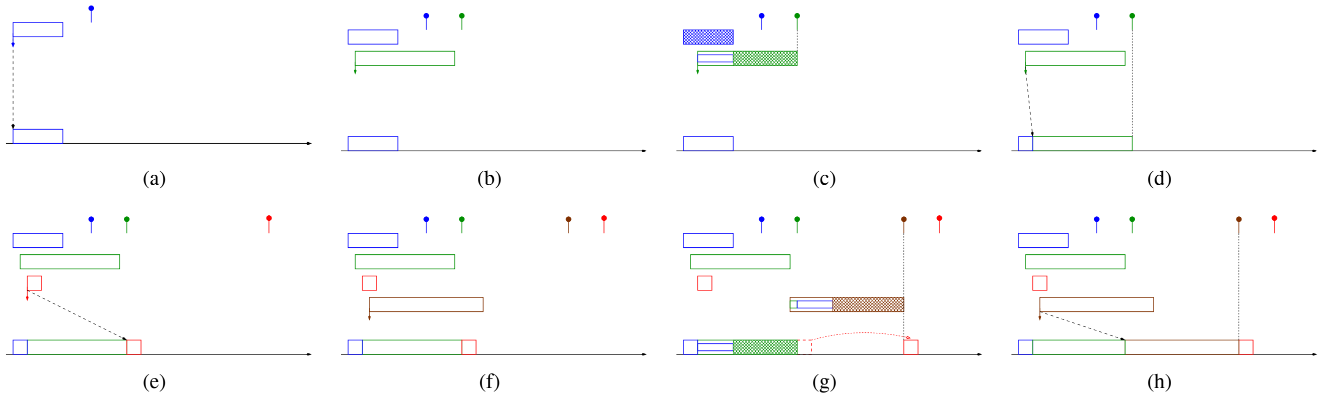
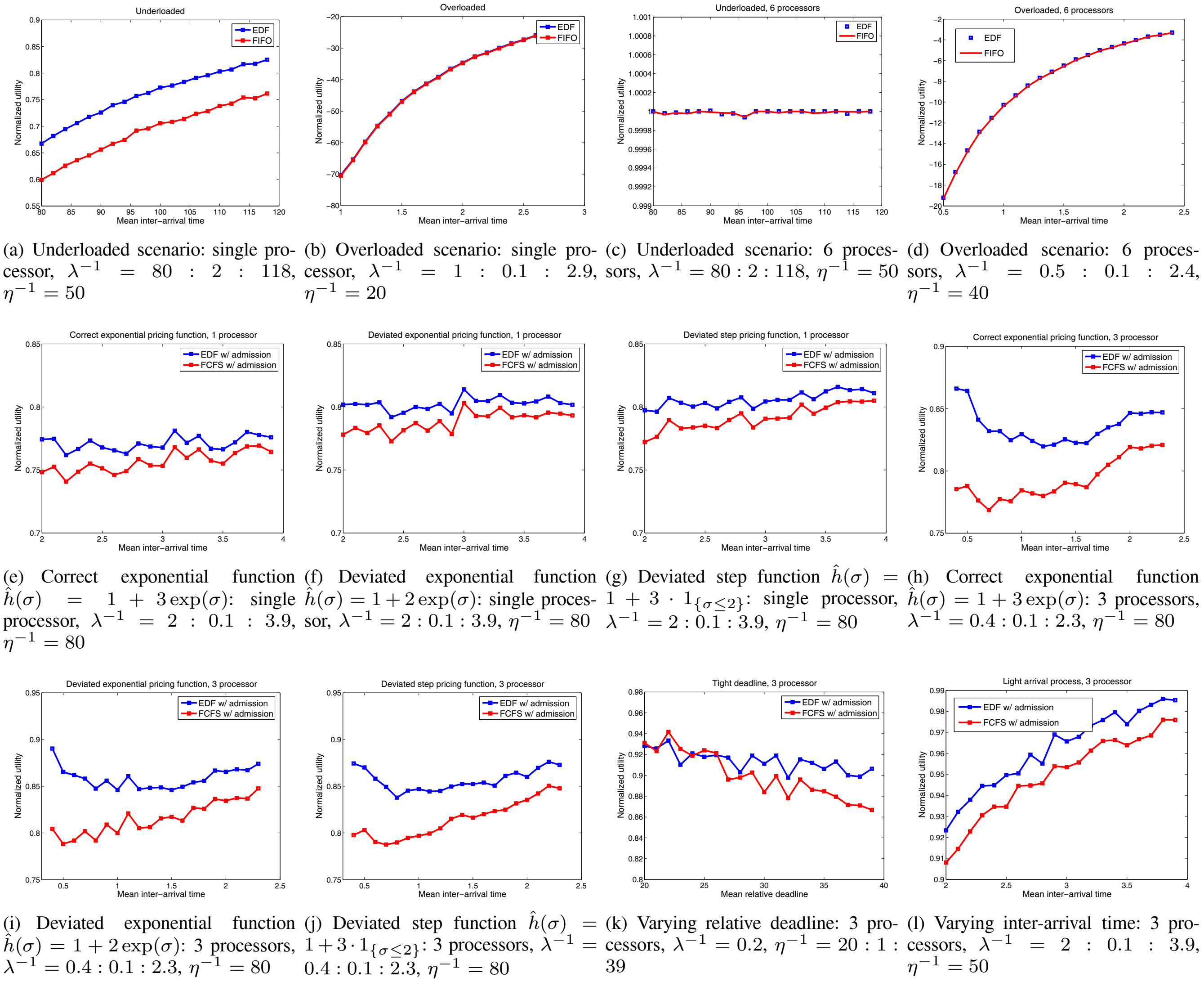


Figure 1. Illustration of the admission and scheduling algorithm



(a) Underloaded scenario: single processor, $\lambda^{-1} = 80 : 2 : 118$, $\eta^{-1} = 50$

(b) Overloaded scenario: single processor, $\lambda^{-1} = 1 : 0.1 : 2.9$, $\eta^{-1} = 20$

(c) Underloaded scenario: 6 processors, $\lambda^{-1} = 80 : 2 : 118$, $\eta^{-1} = 50$

(d) Overloaded scenario: 6 processors, $\lambda^{-1} = 0.5 : 0.1 : 2.4$, $\eta^{-1} = 40$

(e) Correct exponential function $\hat{h}(\sigma) = 1 + 3\exp(\sigma)$: single processor, $\lambda^{-1} = 2 : 0.1 : 3.9$, $\eta^{-1} = 80$

(f) Deviated exponential function $\hat{h}(\sigma) = 1 + 2\exp(\sigma)$: single processor, $\lambda^{-1} = 2 : 0.1 : 3.9$, $\eta^{-1} = 80$

(g) Deviated step function $\hat{h}(\sigma) = 1 + 3 \cdot 1_{\{\sigma \leq 2\}}$: single processor, $\lambda^{-1} = 2 : 0.1 : 3.9$, $\eta^{-1} = 80$

(h) Correct exponential function $\hat{h}(\sigma) = 1 + 3\exp(\sigma)$: 3 processors, $\lambda^{-1} = 0.4 : 0.1 : 2.3$, $\eta^{-1} = 80$

(i) Deviated exponential function $\hat{h}(\sigma) = 1 + 2\exp(\sigma)$: 3 processors, $\lambda^{-1} = 0.4 : 0.1 : 2.3$, $\eta^{-1} = 80$

(j) Deviated step function $\hat{h}(\sigma) = 1 + 3 \cdot 1_{\{\sigma \leq 2\}}$: 3 processors, $\lambda^{-1} = 0.4 : 0.1 : 2.3$, $\eta^{-1} = 80$

(k) Varying relative deadline: 3 processors, $\lambda^{-1} = 0.2$, $\eta^{-1} = 20 : 1 : 39$

(l) Varying inter-arrival time: 3 processors, $\lambda^{-1} = 2 : 0.1 : 3.9$, $\eta^{-1} = 50$

Figure 2. Illustration of the performance of the pricing, admission and scheduling algorithm

simulation validates the satisfactory performance of DSAC in both underloaded and overloaded scenarios.

C. Pricing and Utility

Different pricing functions are simulated and the impact of deviation of the pricing function from the true utility function is investigated with respect to the maximization of collective utility. It can be observed that EDF and FCFS with admission control achieve less fraction of the value of DSAC in Fig. 2(e) compared with Fig. 2(f) and Fig. 2(g). Same observation can be made from the 3-processor situation (Fig. 2(h), 2(i) and 2(j)). Since EDF and FCFS with admission control do not adapt to the prices, a smaller ratio of the value of EDF and FCFS over that of DSAC indicates better performance of DSAC in terms of maximizing collective utility.

The effect of the pricing ingredient is thus illustrated by the fact that when the pricing function approximates the true utility function better, the LSC operations achieve better collective utility. It can be observed that moderate deviation in the pricing function ($\hat{h}(\sigma) = 1 + 2\exp(\sigma)$ and $\hat{h}(\sigma) = 1 + 3 \cdot 1_{\{\sigma \leq 2\}}$) from the true utility function ($h(\sigma) = 1 + 3\exp(\sigma)$) does not cause significant degradation in the collective customer utility.

D. Scheduling Impact

The normalized utility is plotted versus the mean inter-arrival time and mean relative deadline, and the impact of the scheduling ingredient is investigated by comparing with EDF and FCFS with admission control. It can be observed in Fig. 2(k) and 2(l) that DSAC performs stronger compared with EDF and FCFS with admission control as the relative deadline gets looser and the inter-arrival time gets smaller.

The effect of the scheduling ingredient can be illustrated by the fact that the benefit associated with DSAC improves when the arrival process gets heavier, indicating the capability of DSAC handling extremely heavy request arrival. Similarly, the benefit associated with DSAC improves when the deadline is relaxed, indicating the capability of DSAC earning increased revenue with customer flexibility.

VI. Conclusion

The problem of EV garage charging management is considered where EV customers arrive sporadically with deadlines and battery charging level requirement. The pricing, admission and scheduling aspects of the large scale charging facility operations are investigated, and a utility based pricing scheme is proposed that explores the customers' time flexibility, together with an online admission and scheduling algorithm DSAC with worst case competitive ratio guarantee for linear utility function. Satisfactory average performance of DSAC is demonstrated for both underloaded and overloaded scenarios via extensive comparative simulation with benchmark algorithms such as EDF and FCFS under a stochastic setup.

References

[1] M. J. Scott, M. Kintner-Meyer, D. B. Elliot, and W. M. Warwick, "Impact assessment of plug-in hybrid vehicles on electric utilities and regional us power grids:, part 2: Economic assessment," in *10th Annual EUEC Conference*, Tucson, AZ, USA, 2007, pp. 1–19.

[2] H. Turton and F. Moura, "Vehicle-to-grid systems for sustainable development: An integrated energy analysis," *Technological Forecasting and Social Change*, vol. 75, pp. 1091–1108, 2008.

[3] S. Chen, T. He, and L. Tong, "Optimal deadline scheduling with commitment," in *2011 Allerton Conference on Communication, Control and Computing*, Oct 2011.

[4] P. Denholm and W. Short, "An evaluation of utility system impacts and benefits of optimally dispatched plug-in hybrid electric vehicles," Tech. Rep., 2006, national Renewable Energy Laboratory.

[5] S. W. Hadley, "Evaluating the impact of plug-in hybrid electric vehicles on regional electricity supplies," in *Bulk Power System Dynamics and Control*, Charleston, SC, USA, 2007.

[6] "Electrification of the transportation system," Tech. Rep., April 2010, in MIT Energy Initiative Symposium. [Online]. Available: http://web.mit.edu/mitei/docs/reports/electrification-transportation-system.pdf

[7] K. Schneider, C. Gerkensmeyer, M. Kintner-Meyer, and R. Fletcher, "Impact assessment of plug-in hybrid vehicles on pacific northwest distribution systems," in *IEEE Power and Energy Society 2008 General Meeting*, Pittsburgh, PA, USA, 2008, pp. 1–6.

[8] C. Roe, J. Meisel, F. Evangelos, T. Overbye, and A. P. Meliopoulos, "Power system level impacts of phevs," in *42nd Hawaii International Conference on System Sciences*, 2009, pp. 1–10.

[9] X. Yu, "Impacts assessment of phev charge profiles on generation expansion using national energy modeling system," in *IEEE Power and Energy Society 2008*, Pittsburgh, PA, USA, 2008.

[10] P. Tulpule, V. Marano, S. Yurkovich, and G. Rizzoni, "Energy economic analysis of pv based charging station at workplace parking garage," in *IEEE EnergyTech 2011*, Cleveland, OH, USA, May 2011, pp. 1–6.

[11] P. Kulshrestha, L. Wang, M.-Y. Chow, and S. Lukic, "Intelligent energy management system simulator for phevs at municipal parking deck in a smart grid environment," in *IEEE Power and Energy Society General Meeting 2009*, Calgary, AB, Canada, July 2009, pp. 1–6.

[12] W. Su and M.-Y. Chow, "Performance evaluation of a phev parking station using particle swarm optimization," in *IEEE Power and Energy Society General Meeting 2011*, Detroit, MI, USA, July 2011.

[13] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of ACM*, vol. 20, pp. 46–61, 1973.

[14] M. Dertouzos, "Control robotics: the procedural control of physical processes," in *International Federation for Information Processing Congress 1974*, Stockholm, Sweden, August 1974, pp. 807–813.

[15] A. Mok, "Fundamental design problems of distributed systems for the hard real-time environment," Ph.D. dissertation, MIT, 1983.

[16] C. D. Locke, "Best-effort decision-making for real-time scheduling," Ph.D. dissertation, CMU, 1986.

[17] S. Baruah, G. Koren, D. Mao, B. Mishra, A. Raghunathan, L. Rosier, D. Shasha, and F. Wang, "On the competitiveness of on-line real-time task scheduling," *Real-Time Systems*, vol. 4, pp. 125–144, 1992.

[18] G. Koren and D. Shasha, "Dover: An optimal on-line scheduling algorithm for overloaded uniprocessor real-time systems," *SIAM Journal of Computing*, vol. 24, pp. 318–339, 1995.

[19] A. Bar-Noy, J. A. Garay, and A. Herzberg, "Sharing video on demand," *Discrete Applied Mathematics*, vol. 129, no. 1, pp. 3–30, 2003.

[20] M. Goldwasser and B. Kerbikov, "Admission control with immediate notification," *Journal of Scheduling*, vol. 6, pp. 269–285, 2003.

[21] J. Ding and G. Zhang, "Online scheduling with hard deadlines on parallel machines," in *2nd International conference on Algorithmic Aspects in Information adn Management*, vol. 4041, Hong Kong, China, 2006, pp. 32–42.

[22] J. Ding, T. Ebenlendr, J. Sgall, and G. Zhang, "Online scheduling of equal-length jobs on parallel machines," in *15th Annual European Symposium on Algorithm*, vol. 4698, 2007, pp. 427–438.

[23] T. Ebenlendr and J. Sgall, "A lower bound for scheduling of unit jobs with immediate decision on parallel machines," in *6th Workshop on Approximation and Online Algorithms*, 2008, pp. 43–52.

[24] N. Thibault and C. Laforest, "Online time constrained scheduling with penalties," in *23rd IEEE International Symposium on Parallel and Distributed Processing*, 2009.

[25] S. P. Y. Fung, "Online preemptive scheduling with immediate decision or notification and penalties," in *16th Annual International Conference on Computing and Combinatorics*, 2010, pp. 389–398.

[26] S. Chen, Y. Ji, and L. Tong, "Deadline scheduling with admission control," in preparation.