Online Welfare Maximization for Electric Vehicle Charging with Electricity Cost

Zizhan Zheng and Ness B. Shroff
Department of Electrical and Computer Engineering
The Ohio State University
{zheng.497, shroff.11}@osu.edu

ABSTRACT

The accelerated adoption of EVs in the last few years has raised concerns that the power grid can get overloaded when a large number of EVs are charged simultaneously. A promising direction is to implement large scale automated scheduling of EV charging at public facilities, by exploiting the time elasticity of charging requests. In this work, we study the problem of online EV charging for maximizing the total value of served vehicles minus the energy cost incurred. In contrast to most previous works that assume a fixed capacity constraint while ignoring the electricity cost, we adopt a convex cost model for the system operator together with a concave valuation model for the vehicle owners. We design an online algorithm for balancing the two and prove a bound on its competitive performance for a general class of valuation and cost functions.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Scheduling; I.1.2 [**Algorithms**]: Analysis of algorithms

Keywords

Electric vehicle charging; deferrable load control; online algorithms

1. INTRODUCTION

The past few years have witnessed increasing interest in Electrical Vehicles (EVs) including both plug-in hybrid electric vehicles (PHEVs) and fully electric vehicles, driven by the advances in battery technology and the necessity of reducing carbon emissions and dependence on petroleum. It is projected that the adoption of EVs is likely to accelerate in the next decade. For instance, the U.S. government calls for deploying 1 million PHEVs by 2015 [17], and a recent Gartner report estimates that by 2020, 10% of all vehicle sales will be EVs [13].

e-Energy'14, June 11–13, 2014, Cambridge, UK.

http://dx.doi.org/10.1145/2602044.2602053.

The accelerated adoption of EVs, however, leads to the concern that when a large number of EVs are charged simultaneously in a local area, which is likely to happen in the near future, the local power grid can easily get overloaded. To address the problem, a promising direction is to investigate large scale automated scheduling of EV charging [1, 8, 10]. The key observation is that vehicle owners often exhibit some flexibility in their charging requests, including the time period of getting charged, and in the case of PHEVs, the total amount as well. By exploiting the statistical multiplexing gain and the time elasticity of charging requests, coordinated charging can greatly improve energy efficiency while meeting the utility of vehicle owners. Such a scheme can be implemented in public areas such as parking garages and working places, as envisioned in [8].

In this work, we study the problem of online EV charging for maximizing the total value of served vehicles minus the energy cost incurred. As in [10, 18, 19], each request is characterized by a time window that models the time elasticity of the vehicle owner, a concave function that models the non-increasing marginal valuation for incremental unit of electricity, and a charging rate limit. Moreover, we model the energy cost at any time as a convex function of the total load at that time. The convex cost model has been widely adopted for modeling energy cost, and reflects the fact that each additional unit of power for meeting the increased load is more expensive to obtain [14, 16].

We have designed an online algorithm to this problem that requires no knowledge of future requests while achieving a comparable efficiency as the optimal offline solution. Our analysis is built upon a recently developed primal-dual framework for competitive analysis of online algorithms [6, 12], while allowing a more general class of utility and cost functions. In addition to EV charging, our study applies to welfare maximization of other types of resources where the demand side has a concave utility and exhibits time elasticity, while the supply side incurs a convex cost, e.g., scheduling of computing tasks in a data center and allocating bandwidth in a communication network.

There are several online algorithms designed for coordinated EV charging [7,8,10,19]. In [10], a greedy algorithm is proposed for maximizing the total valuations of vehicle owners subject to a capacity constraint of the distribution network. To cope with the strategic behavior of selfish agents, the algorithm is extended to an online mechanism by allowing some pre-allocated units to be "burned." The approach is further extended in [19] to allow multiple charging rates. In another direction, the problem of EV charging with com-

mitment is considered in [7, 8], where each agent requests a fixed amount of resource, and a request has to be either accepted or rejected at its arrival time. A penalty is incurred if an accepted request is not fulfilled by the deadline.

However, most previous works on EV charging assume a capacity constraint while ignoring the electricity cost. One exception is [8], where a linear cost is considered together with a linear valuation. For a large system with high peak load, however, it is often more expensive to generate the supplementary power for meeting the peak load, and a convex cost model better reflects the real cost of electricity [14, 16]. On the other hand, the problem of minimizing a convex energy cost for serving deferrable electric load has also been considered, under the assumption that all the requests have to be satisfied in full [14]. In practice, however, the charging demand of an EV often exhibits some flexibility in terms of the total amount needed, which is better captured by a concave valuation function. Moreover, the approach in [14] does not provide any worst-case performance guarantee. Our approach generalizes these two extreme cases by modelling flexibilities in both the demand and the supply, a better reflection of the reality, while ensuring a performance bound even in the worst-case. In the offline setting, welfare maximization for electrical load management with general concave valuations and convex costs has been considered in [16]. However, the problem has not been studied in the *online* setting to the best of our knowledge. In a different context, a recent work considered the problem of pricing a set of items with general production cost to a sequence of buyers to maximize social welfare or profit [3]. The approach does not apply to expiring resources like electricity and does not consider the time elasticity of demand.

Our main contribution can be summarized as follows.

- We develop an online algorithm that balances the total value of served vehicles and the total energy cost for charging.
- For continuous charging rate (and under some further assumptions to be defined precisely in Section 4), we establish a performance bound of our online algorithm compared with the optimal offline algorithm, based on a characterization of the concavity (resp. convexity) of the valuation functions (resp. cost functions). We further study the competitive performance of our algorithm for concrete examples of valuation and cost functions.
- Simulation results show that our algorithm achieves close to optimal performance even when the charging rate is discrete, compared with the optimal offline solution with continuous charging rate.

The remainder of this paper is organized as follows. We present the system model and the welfare optimization problem in Section 2. Our online algorithm is developed in Section 3, where we also present two heuristics as baselines. The competitive performance of our algorithm is studied in Section 4. We evaluate our algorithm in Section 5, and conclude the paper in Section 6.

2. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we discuss our system model and major assumptions made, and present the optimization problem to be studied.

Consider a system operator that manages multiple charging points. As in [10,19,20], we assume that there are enough charging points so that no agent needs to wait to be charged. A time-slotted system is considered. In any time-slot t, the operator incurs a cost $g_t(z_t)$ for serving a total amount z_t (kWh) of charging request. Note that $g_t(\cdot)$ may vary over t in general. We assume that $g_t(\cdot)$ is non-decreasing and convex, $g_t(0) = 0$, and $g_t(\cdot)$ is known to the operator [1]. Let $\mathcal{N} = \{1, ..., N\}$ denote a set of agents, each operating a single EV on behalf of its owner. Each agent i is described by its *type* $\theta_i = (f_i, a_i, d_i, X_i, Y_i)$, where $f_i(\cdot)$ specifies its valuation function, X_i denotes its maximum charging rate, i.e., the maximum amount of electricity that i can charge in any time-slot, and Y_i is the maximum amount of electricity that i would like to obtain, beyond which, there is no extra value. Agent i arrives at the beginning of time-slot a_i, and departs by the end of time-slot d_i, and obtains a valuation $f_i(y_i)$ if it receives a total amount of electricity y_i on departure. Such a total energy requirement model with continuous service rate has been used in [1, 9] for modeling energy demand of EVs and other deferrable electric loads. As in [19], we assume that $f_i(\cdot)$ is non-decreasing and concave, and $f_i(0) = 0$ for any i. Note that Y_i can be incorporated into the definition of f_i. We choose to separate them for the sake of clarity. We assume that the agents always report their true types on their arrivals. Extension of our online algorithm to a truthful online mechanism will be part of our future study. Let $D = \max_i(d_i - a_i + 1)$ denote the maximum time elasticity of any requests.

Agents are sorted by their arrival times (ties are broken arbitrarily). Let T denote the last departure time, i.e., $T = \max_{i\in\mathcal{N}} d_i$. Let x_{it} denote the amount of electricity that agent i obtains at time t, and let $\mathbf{x} \triangleq \{x_{it} : 1 \le i \le N, 1 \le t \le T\}$. Our objective is to maximize the social welfare of both the system operator and the customers, that is, the total valuations of customers minus the total electricity cost:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{i=1}^{N} f_i\Big(\sum_{t=1}^{T} x_{it}\Big) - \sum_{t=1}^{T} g_t\Big(\sum_{i=1}^{N} x_{it}\Big) \\ \text{s.t.} \quad & 0 \le x_{it} \le X_i, && \forall i, t, \\ & x_{it} = 0, && \forall i, t \notin [a_i, d_i], \\ & \sum_{t=1}^{T} x_{it} \le Y_i, && \forall i. \end{aligned}$$

We note that the offline problem is a convex optimization problem, which can be solved to any accuracy in polynomial time for a large class of f and g. Our objective, however, is to study the problem in the more challenging online setting, where agents of different types arrive on the fly, and at any time t, the system operator only has the information of agents that are currently in the system and that have left the system by t. Our objective is to design online algorithms that are competitive with respect to the optimal offline algorithm. An online algorithm is q-competitive for some $q \ge 1$ if it achieves at least $1/q$ of the optimal offline social welfare in the worst case [4].

As we will discuss in Section 4, our analysis focuses on the "continuous" setting when f_i and g_t are continuously differentiable, and x_{it} are continuous. However, our algo-

[1]At any time t, it is sufficient for our algorithm to know $g_\tau(\cdot)$ for $\tau \in \{t, t+1, ..., t+D-1\}$, where $D = \max_i(d_i - a_i + 1)$.

Table 1: Notation List

$\mathcal{N}$	Set of agents (charging requests)
N	Total number of requests
T	Last departure time
g_t	Cost function at time t
$c_{t,k}$	Marginal cost for serving the k-th unit at time t
z_t	Total amount of electricity consumed at time t
f_i	Valuation function of agent i
$v_{i,k}$	Marginal valuation of the k-th unit for agent i
a_i, d_i	Arrival time, departure time of agent i
θ_i	Type of agent i, where $\theta_i = (f_i, a_i, d_i, X_i, Y_i)$
D	$\max_{i\in\mathcal{N}}(d_i - a_i + 1)$
x_{it}	Amount of electricity given to agent i at time t
y_i	Total amount of electricity given to agent i
X_i	Maximum charging rate of agent i
Y_i	Maximum amount of electricity required by agent i
δ	Charging unit
γ	Revocation coefficient

rithm applies to more general forms of f_i and g_t, and to the "discrete" case when the values of x_{it} need to be a multiple of some charging unit δ. In the discrete case, we define $v_{i,k} = f_i(k\delta) - f_i((k-1)\delta)$ as the k-th *marginal valuation* to agent i. Then we have $f_i(y_i) = \sum_{k=1}^{\lfloor y_i/\delta \rfloor} v_{i,k}$. The concavity of f_i implies that $v_{i,k} \geq v_{i,k+1}, \forall i, k$. Hence, a discrete valuation function can be equivalently defined as a vector of marginal valuations. Similarly, we define $c_{t,k}$ as the k-th marginal cost for the cost function g_t. The convexity of g_t then implies that $c_{t,k} \leq c_{t,k+1}, \forall t, k$. Note that the continuous case can be viewed as an extreme of the discrete case when $\delta \to 0$. To simplify the notation, we also use marginal valuation (cost) to denote the derivative of a continuous valuation (cost) function. Simulation results show that our algorithm achieves close to optimal performance even when x_{it} are discrete, compared with the continuous offline optimal (note that in general, it is NP-hard to find the optimal (offline) solution in the discrete case).

In our analysis and evaluation, we will consider the following commonly adopted valuation and cost functions as examples (our analysis applies to more general f and g). For cost functions, we will consider $g_t(z_t) = c_t z_t^\alpha$ for some $c_t > 0$ and $\alpha \geq 1$, where c_t varies over time, which has been widely adopted for modeling energy cost [14]. We also consider two extensions to this model that incorporate free renewable energy and base load, respectively (see Section 4.3). For valuation functions, we will consider $f_i^1(y_i) = v_i \log(1 + y_i)$ and $f_i^2(y_i) = v_i y_i^\beta$ for $v_i > 0$ and $\beta \in [0, 1]$. Note that when the parameter v_i is identical for all the agents, f_1 and f_2 are closely related to the well-known notions of proportional fairness and α-fairness [15], respectively.

3. ONLINE ALGORITHMS

In this section, we present our online algorithms for the EV charging problem. We first consider the case when f and g are continuously differentiable, and x_{it} are continuous. Extensions to more general f and g and to discrete x_{it} will be discussed at the end of the section. We start with two simple online solutions adapted from existing algorithms proposed in related settings [12, 19] (Section 3.1). A careful study of these algorithms reveals their weakness in our context. They also serve as baselines in our simulations (see Section 5). We then propose a more sophisticated solution in 3.2.

3.1 Two Simple Algorithms

The first algorithm we consider is adapted from an online algorithm for a multi-speed EV charging problem proposed in [19], where instead of the electricity cost, a capacity constraint is considered. The objective is to maximize the total valuations of all the agents subject to a constraint on the total amount that can be served at any time. For any agent i, let y_{it} denote the total amount of electricity that agent i received by time t. An agent i is *active* at time t if $t \in [a_i, d_i]$ and $y_{it} < Y_i$. The online algorithm greedily serves the active requests in each time-slot as follows. At any time t, active agents are served in a non-decreasing order of $f_i'(y_{it})$ subject to the capacity constraint and the charging rate limit of each agent. Let z_t denote the current load at time t. To adapt this algorithm to our problem, we observe that when electricity cost is introduced, it is beneficial to serve agent i at time t only if $f_i'(y_{it}) > g_t'(z_t)$. We therefore modify the algorithm as follows.

Per-Time Allocation (PT): At each time t, active agents are served in a non-decreasing order of their marginal valuations, subject to their charging rate constraint. The process repeats until $\max_i f_i'(y_i) \leq g_t'(z_t)$.

Note that PT does not exploit time elasticity explicitly. As an illustrative example, consider a system of three agents, and assume $g_t(z_t) = 0.5z_t^2$ for all t. Agent 1 and 2 arrive at time 1, where $\theta_1 = (1.5y_1, 1, 2, 1, 1)$ and $\theta_2 = (1.5y_2, 1, 1, 1, 1)$ (see Figure 1 (a)). Since the two agents have the same marginal valuation, they can be served in any order. Assuming they are served with equal chance, each of them receives 0.75 units in the first time-slot, so that the marginal valuation of each agent equals to the marginal cost in the first time-slot, both equal to 1.5. Agent 2 then departs at the end of time-slot 1 and agent 3 arrives at the beginning at time-slot 2 where $\theta_3 = (2y_3, 2, 3, 2, 3)$. Note that at time 2, agent 3 receives 2 units while agent 1 (still in the system) is not served as it has lower marginal valuation than agent 3. Finally, at time 3, agent 3 receives 1 more unit since $Y_3 = 3$ and then departs. The total welfare achieved can be computed as $1.5 \times (0.75 + 0.75) + 2 \times 3 - 0.5 \times (1.5^2 + 2^2 + 1^2) = 4.625$.

The second algorithm that we consider is originally proposed for serving computing tasks using a single machine with speed scaling [21]. In this context, each job requires certain amount of CPU cycles, and the power consumption at any time is a function of the processor speed. The objective is to minimize the total energy consumption for serving all the requests (partial fulfillment is not beneficial) subject to their deadline constraints. We consider the greedy online algorithm recently proposed in [12]. In contrast to PT, a plan is made for each request at its arrival time to exploit the time elasticity of requests. The algorithm can be interpreted in our context as follows. A schedule is found for each request at its arrival time, which remains fixed for its entire lifetime in the system. To serve agent i, the time slots in $[a_i, b_i]$ of minimum load (given the current allocations made) are first considered, subject to the charging rate constraint. The process repeats until i's request is satisfied. The algorithm then moves on to the next agent. A simple approach to adapt this algorithm to our setting is to stop allocation for agent i when $f_i'(y_i) \leq \min_{\tau\in[a_i,b_i]} g_\tau'(z_\tau)$.

On-Arrival Allocation (OA): At each time t, for each agent i that arrives at t, the load of all the time slots $\tau \in [a_i, d_i]$ with minimum $g_\tau'(z_\tau)$ are increased for serving i, sub-

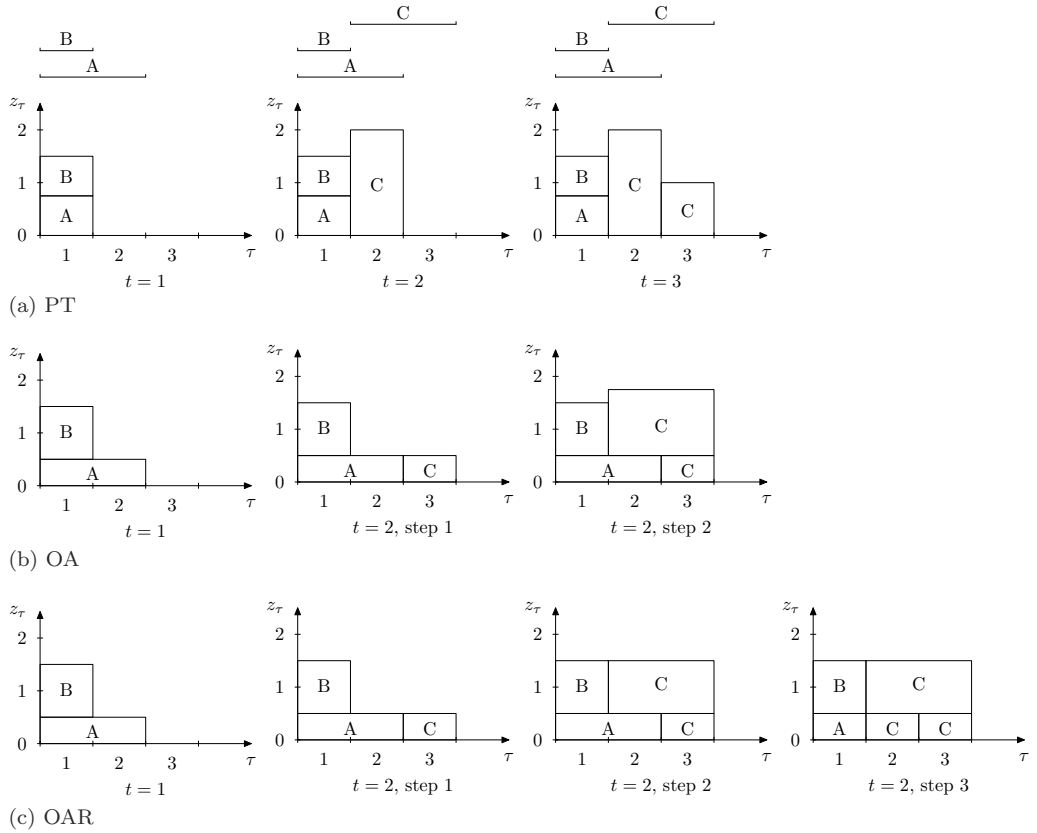


Figure 1: An example of EV charging with three agents.

ject to the charging rate constraint, until $f_i'(y_i) \leq \min_{\tau \in [a_i, b_i]} g_\tau'(z_\tau)$.

Consider the three agent example again. In the beginning of time-slot 1, a plan is made for agent 1 first. Agent 1 receives 0.5 unit in time-slot 1 and 0.5 unit in time-slot 2 (see Figure 1(b)). Agent 2 then receives 1 unit in time-slot 1. When agent 3 arrives at time 2, time-slot 3 is first considered for serving agent 3 as it has the minimum load. Once the load of time 3 increases to 0.5, the load of time-slot 2 increases together with that of time-slot 3 until agent 3 receives its maximum required amount of 3 units, with 1.25 unit served in time 2 and 1.75 unit served in time 3. The total welfare achieved in this case is $1.5 \times (1+1) + 2 \times 3 - 0.5 \times (1.5^2 + 1.75^2 + 1.75^2) = 4.8125$.

Discussion: The main problem with the PT algorithm is that the time elasticity of requests is largely ignored. In particular, the algorithm tends to serve requests as fast as possible subject to the charging rate constraint, which can lead to a high cost. Consider a simple example with m agents of the same type $\theta_i = (y_i, 1, m, 1, 1)$, and $g_t(z_t) = 0.25 z_t^2$ for all t. Then PT will serve all the agents in the first $m/2$ time-slots, with two agents served together in each of these time-slots, leading to a welfare of $m - m/2 = 0.5m$. On the other hand, in the optimal solution, each agent is served 1 unit in a different time-slot, leading to a welfare of $m - 0.25m = 0.75m$. This result can be made worse when the marginal cost is smaller. In general, we expect that PT performs worse when the average time elasticity is high and traffic load varies over time (so that the time elasticity can potentially be exploited), which is confirmed in our simulations. The example reveals that instead of making decisions for each time-slot separately, a plan for the future is needed to reduce the cost.

On the other hand, the OA algorithm tends to reduce electricity cost by making a plan for each agent as early as possible. However, since a schedule for each request is fixed at its arrival time and cannot be modified for its entire lifetime, it can prevent future requests with higher marginal valuations from being served. This can happen especially when the system load is relatively high. For instance, consider the three agent example in Figure 1(b) again. The key observation is that the welfare can be improved if part of the allocation to agent 1 can be revoked to serve agent 3. Note that the decision for agent 3 is made at the beginning of time 2, where the allocation for agent 1 at time 2 can still be modified. In making the scheduling decision for agent 3, when the load of both time-slots 2 and 3 increase to 1.5, the marginal cost in both time slots equals to the marginal valuation of agent 1. Starting from that point, instead of further increasing the load, it is more efficient to displace the allocation made for agent 1 at time 2 and reassign the units to agent 3 (see Figure 1(c)). The total welfare is then improved to $1.5 \times (1+0.5) + 2 \times 3 - 0.5 \times (1.5^2 \times 3) = 4.875$.

3.2 An Online Algorithm with Revocation

Based on the above discussion, we then design an online algorithm that combines two key ideas: (1) for any request, a tentative schedule that looks into the future is needed for exploiting the time elasticity of requests; (2) tentative allocations made for existing requests should be revocable for serving new requests with higher valuations. Our algorithm

is built upon the framework of OA, where for any agent i, a tentative schedule is determined at its arrival time a_i. In making the decision, however, allocations within the interval $[a_i, d_i]$ made for previous agents can be reassigned to agent i if the marginal valuation of the latter is higher than the marginal valuation of the former by a factor $\gamma \geq 1$. The parameter γ is called ***revocation coefficient*** and is selected by the algorithm according to the shapes of the valuation and cost functions. Approaches for determining a desired γ will be studied in Section 4.

We now discuss our On-Arrival Allocation with Revocation (OAR) algorithm in detail. The algorithm maintains the total amount of electricity consumed in each time-slot t, denoted as z_t, which is initialized to 0. At any time t, for each agent i that arrives at t, a tentative schedule is made for i as follows. For each time-slot $\tau \in [a_i, b_i]$, let $x_{i\tau}$ denote the amount of electricity given to i in time τ, and let $y_i = \sum_\tau x_{i\tau}$ denote the total amount that i receives. These variables are initialized to 0. The tentative schedule for agent i is made in multiple iterations (lines 3-41). In each iteration, certain amount of electricity is allocated to i until y_i equals to Y_i, the maximum amount that agent i requires, or when there is no benefit of serving more for i.

In each iteration, let H denote the set of time-slots in $[a_i, d_i]$ where agent i can receive more electricity subject to its charging rate limit (line 4). Let dc denote the minimum marginal cost in these time-slots, and let H' denote the set of time-slots in H that achieves this minimum (line 9). These are the most cost-effective time-slots for serving i. Moreover, let J denote the set of agents from which the current allocation can potentially be revoked for serving i (lines 10-11). These agents have to satisfy the following necessary conditions:

- They arrived before i;
- They have received some amount in H';
- Their marginal valuation (with respect to their current allocations) is less than $f_i'(y_i)/\gamma$, where $\gamma \geq 1$ is the revocation coefficient, and no larger than dc.

If J is nonempty, let dv denote the minimum marginal valuation for agents in J, and let J' denote the set of agents in J that achieves the minimum. Otherwise, let $dv = dc$ and $J' = \emptyset$ (lines 12-18).

If neither dc nor dv is less than $f_i'(y_i)$, there is no benefit of serving more for agent i. The algorithm moves on to the next agent (lines 19-21). Otherwise, we set $H' = \emptyset$ if $dv < dc$, since in this case, it is more cost-effective to revoke existing allocation instead of allocating more. The algorithm then invokes a procedure Increment to identify a small increment δ_τ for each $\tau \in H'$, and a small amount δ_j to be displaced for each $j \in J'$ (line 25). The procedure Increment identifies the maximum possible amount of increment (revocation) while meeting the following criteria:

- The new value of y_i after increment (revocation) is bounded by Y_i;
- The new marginal cost (valuation) of these time-slots (agents) after increment (revocation) should still equal to each other, and no larger than the derivatives of untouched time-slots in H and agents in J, and that of $f_i'(y_i)$;
- If $\delta_j > 0$ is applied to an agent $j \in J'$, then δ_j should be small enough so that $f_j'(y_j) \leq f_i'(y_i)/\gamma$ after revocation.

On-Arrival Allocation with Revocation (OAR)

$z_t \leftarrow 0, \forall t$
$\gamma \leftarrow \gamma^*$ as determined by Equation (12)
In each time-slot t

```
1: for each agent i that arrives at t do
2:     x_iτ ← 0, y_i ← 0, ∀i, τ
3:     while y_i ≤ Y_i do
4:         H ← {τ : a_i ≤ τ ≤ d_i, x_iτ < X_i}
5:         if H == ∅ then
6:             break
7:         end if
8:         dc ← min_{τ∈H} g'_τ(z_τ)
9:         H' ← {τ ∈ H : g'_τ(z_τ) == dc}
10:        J ← {j < i : x_jτ > 0 for some τ ∈ H',
11:              f'_j(y_j) < f'_i(y_i)/γ, f'_j(y_j) ≤ d_c}
12:        if J == ∅ then
13:            dv ← dc
14:            J' ← ∅
15:        else
16:            dv ← min_{j∈J} f'_j(y_j)
17:            J' ← {j ∈ J : f'_j(y_j) == dv}
18:        end if
19:        if f'_i(y_i) ≤ min(dc, dv) then
20:            break
21:        end if
22:        if dv < dc then
23:            H' ← ∅
24:        end if
25:        ({δ_j}, {δ_τ}) ← Increment(H, J, H'J', y_i)
26:        for τ ∈ H' do
27:            z_τ ← z_τ + δ_τ, x_iτ ← x_iτ + δ_τ, y_i ← y_i + δ_τ
28:        end for
29:        for j ∈ J' do
30:            δ_0 ← δ_j
31:            for τ ∈ H' and x_jτ > 0 do
32:                δ_1 ← min(δ_0, x_jτ, X_i − x_iτ)
33:                x_jτ ← x_jτ − δ_1, y_j ← y_j − δ_1
34:                x_iτ ← x_iτ + δ_1, y_i ← y_i + δ_1
35:                δ_0 ← δ_0 − δ_1
36:                if δ_0 ≤ 0 then
37:                    break
38:                end if
39:            end for
40:        end for
41:    end while
42: end for
```

The load in each time-slot $\tau \in H'$ then increases by δ_τ, and the allocation of i is updated accordingly (lines 26-28). Similarly, for each agent $j \in J'$, total amount of δ_j is displaced from the time-slots in H' where j has non-zero allocation, which is reassigned to i subject to the charging rate constraint of i (lines 29-40).

Remark 1: When the functions f and g have constant or linear derivatives, the procedure Increment can be easily implemented to identify the maximum possible increment (revocation) subject to the required conditions. For general f and g, however, it can be difficult to satisfy the criteria above exactly while still making some progress in each step, and some approximation might be needed. We will adopt the following simple solution in our simulations. In each iteration, only consider one time-slot τ in H' or one agent j in J', and increase the load of τ or revoke the allocation of j by a small fixed amount $\delta > 0$, where δ can be adjusted to trade off the accuracy and the time complexity. We call this procedure Simple-Increment.

One advantage of introducing a step size δ is that the algorithm can then be easily extended to non-differentiable f and g, and to the discrete case where x_{it} requires to be a multiple of a charging unit (a given system parameter). In the later case, we can simply set δ to be the charging unit, and replace all the derivatives by marginal valuations or marginal costs. An extra unit of agent i is served at time τ only if the marginal valuation of the next unit of i is higher than the marginal cost for serving one more unit at time τ. Similarly, a unit of agent j is displaced by a unit of agent i only if the marginal valuation of the next unit of i is higher than the marginal valuation of the last unit of j.

Remark 2: With Simple-Increment applied, the time complexity of the OAR algorithm can be determined as follows. Let N denote the number of agents, $Y = \max_i Y_i$ the maximum battery size of an EV, and $D = \max_i(d_i - a_i + 1)$ the maximum time elasticity of any agent. Each agent then requires $O(Y/\delta)$ iterations to schedule, and the time complexity of one iteration is dominated by computing the set J (lines 10-11), which requires $O(ND)$ time. Therefore, the algorithm has a time complexity of $O(\frac{Y}{\delta}N^2D)$.

4. ANALYSIS OF ONLINE ALGORITHMS

We next study the performance of our online algorithm presented in the previous section, by adopting a primal-dual framework [6, 12]. As a classic tool for the design and analysis of approximation algorithms in the offline setting, primal-dual approach has recently been successfully applied to online optimization with linear objectives [5, 6]. More recently, this approach has been extended to online non-linear optimization as well. In particular, it has been used in [12] to prove competitive results for the online energy minimization algorithm in the special case when the cost function is a power function, as we mentioned in Section 3.1. Our analysis extends this approach by considering both a general convex cost and a general concave valuation function. To strike a balance between the two, our approach is to properly determine the value of the revocation coefficient γ according to the shapes of the valuation and cost functions. In our proof, we need the following assumption (in addition to the assumptions made in Section 2).

ASSUMPTION 4.1. *f and g are continuously differentiable and strictly increasing; x_{it} are continuous (and can be arbitrarily small); and the Increment procedure (discussed in Section 3.2) can by implemented exactly.*

As we mentioned in Section 3.2, our algorithm applies to the more general setting where a discrete charging unit δ can be introduced. We expect that, when the unit δ can be made small enough[2], similar competitive results as the ones established below can be obtained for more general f and g, e.g., a piecewise linear concave or convex function, with little loss, at the expense of a higher complexity of the algorithm. Extending our results to the setting where δ is a given system parameter is part of our future work. On the other hand, we observe in our simulations (see Section 5) that our algorithm achieves close to optimal performance even under the discrete setting.

[2]This can be made more precisely as follows. Consider a system with a single request i and a single time-slot t, where $a_i \le t \le d_i$. Let C_{it} denote the optimal amount of request i scheduled at time t that maximizes the welfare (ignoring other requests and other time slots). Let $C = \min_{i,t} C_{it}$. Effectively, C can be viewed as a notion of the capacity of the system. Then we require that $\delta \ll C$.

4.1 Preliminaries

To illustrate the primal-dual approach, we first rewrite the primal problem as follows:

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{y}} \quad & F(\mathbf{x},\mathbf{y}) = \sum_i f_i(y_i) - \sum_t g_t(\sum_i x_{it}) \\
\text{s.t.} \quad & y_i \le \sum_t x_{it}, && \forall i, && (1)\\
& x_{it} \le X_i, && \forall i,t, && (2)\\
& x_{it} = 0, && \forall i, t \notin [a_i, d_i], && (3)\\
& x_{it} \ge 0, && \forall i,t, && (4)\\
& 0 \le y_i \le Y_i, && \forall i. && (5)
\end{aligned}$$

where $\mathbf{x} \triangleq \{x_{it}\}$ and $\mathbf{y} \triangleq \{y_i\}$.

We introduce a dual variable λ_i for the first constraint for each i, and a dual variable μ_{it} for the second constraint for each i and t. Let $\lambda \triangleq \{\lambda_i\}$ and $\mu \triangleq \{\mu_{it}\}$. Let $\mathcal{X}$ denote the set of $\mathbf{x}$ that satisfies constraints (3) and (4), and let $\mathcal{Y}$ denote the set of $\mathbf{y}$ that satisfies the last constraint. We consider the following dual function

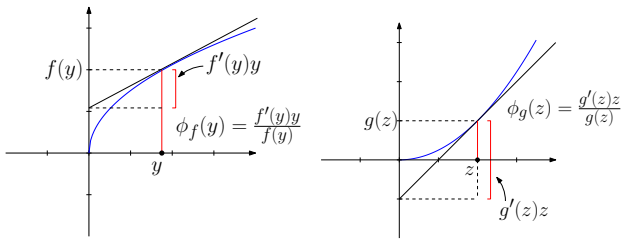
$$\begin{aligned}
G(\lambda,\mu) &= \max_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} \sum_i f_i(y_i) - \sum_t g_t(\sum_i x_{it}) \\
&\qquad + \sum_i \lambda_i(\sum_t x_{it} - y_i) + \sum_{i,t} \mu_{it}(X_i - x_{it}) \\
&= \max_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} \sum_i (f_i(y_i) - \lambda_i y_i) + \sum_{i,t} \mu_{it} X_i \\
&\qquad + \sum_t \Big[\sum_i (\lambda_i - \mu_{it}) x_{it} - g_t(\sum_i x_{it}) \Big]
\end{aligned} \tag{6}$$

By the weak duality theorem [2], the dual function yields an upper bound on the optimal solution of the initial problem for any $\lambda_i \ge 0$ and $\mu_{it} \ge 0, \forall i,t$. The main idea of the online primal-dual approach is to set dual variables $(\widehat{\lambda}, \widehat{\mu})$ based on the values of the primal variables $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ determined by a (deterministic) online algorithm such that $G(\widehat{\lambda}, \widehat{\mu}) \le qF(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ for some $q \ge 1$, which then implies that the online algorithm is q-competitive.

4.2 Analysis

We now study the performance of OAR. Our analysis is centered at the choice of revocation coefficient γ based on a characterization on the level of concavity (resp. convexity) of f (resp. g). By the concavity of f and the assumption that $f(0) = 0, f(y) \ge 0, \forall y \ge 0$, we have $f'(y)y \le f(y)$ for any $y \ge 0$ (a formal proof can be found in [2]). Based on this observation, we use $\phi_f(y) = \frac{f'(y)y}{f(y)}$ to characterize the convexity of f at point y (see Figure 2 for an example), and define $\phi_f = \max_{y\ge 0} \phi_f(y)$. For instance, when $f(y) = vy^\beta$ for some $v > 0, \beta \le 1$, we have $\phi_f = \beta$. Similarly, we consider $\phi_g(z) = \frac{g'(z)z}{g(z)}$ as a characterization of the convexity of g at point z, and define $\phi_g = \min_{z\ge 0} \phi_g(z)$. Moreover, for a given set of agents $\mathcal{N}$ and a time horizon T, we define $\Phi_f = \max_{i\in\mathcal{N}} \phi_{f_i}, \Phi_g = \min_{t\in T} \phi_{g_t}$.

Consider an agent i scheduled by OAR. Let $\{\widetilde{x}_{it}\}$ denote the initial allocation made for i at its arrival, $\widetilde{y}_i = \sum_t \widetilde{x}_{it}$



(a) ϕ_f (b) ϕ_g

Figure 2: Characterization of concavity (convexity) by ϕ_f (ϕ_g).

the total amount of electricity given to i initially, and $\widetilde{z}_{i,t}$ the load in time-slot t right after the initial decision for i is made. Let $\{\overline{x}_{it}\}$, $\overline{y}_i$, and $\overline{z}_t$ denote the corresponding values in the final allocation. We then set the dual variables as

$$\widehat{\lambda}_i = \begin{cases} \max_{\tau:\widetilde{x}_{i\tau}>0} g'_\tau(\widetilde{z}_{i,\tau}) & \text{if } \widetilde{y}_i = Y_i, \\ f'_i(\widetilde{y}_i) & \text{if } \widetilde{y}_i < Y_i. \end{cases}$$

$$\widehat{\mu}_{it} = \begin{cases} \widehat{\lambda}_i - g'_t(\widetilde{z}_{i,t}) & \text{if } \widetilde{x}_{it} > 0, \\ 0 & \text{if } \widetilde{x}_{it} = 0. \end{cases}$$

That is, if $\widetilde{y}_i = Y_i$, $\widehat{\lambda}_i$ is set to the maximum marginal cost for time slots where i has been initially allocated some amount. Otherwise, $\widehat{\lambda}_i$ is set to the derivative of f_i at $\widetilde{y}_i$. Moreover, $\widehat{\mu}_{it}$ satisfies the following complementary slackness condition.

LEMMA 4.1. *$\widehat{\mu}_{it} = 0$ when $\widetilde{x}_{it} < X_i$.*

PROOF. For any agent i and time-slot t, if $\widetilde{x}_{it} = 0$, then $\widehat{\mu}_{it} = 0$ by the definition. Assume $\widetilde{x}_{it} > 0$. We need to show that $\widehat{\lambda}_i = g'_t(\widetilde{z}_{i,t})$. First note that when agent i is initially scheduled in OAR, time-slots with minimum marginal cost are always considered first. Therefore, we must have $g'_t(\widetilde{z}_{i,t}) = \max_{\tau:\widetilde{x}_{i\tau}>0} g'_\tau(\widetilde{z}_{i,\tau})$ by the assumption that $\widetilde{x}_{it} < X_i$. If $\widetilde{y}_i = Y_i$, we then have $\widehat{\lambda}_i = \max_{\tau:\widetilde{x}_{i\tau}>0} g'_\tau(\widetilde{z}_{i,\tau}) = g'_t(\widetilde{z}_{i,t})$. Next consider the case $\widetilde{y} < Y_i$. We must have $f'_i(\widetilde{y}_i) \geq g'_t(\widetilde{z}_{i,t})$ since $\widetilde{x}_{it} > 0$. Moreover, since $\widetilde{x}_{it} < X_i$, we must have $f'_i(\widetilde{y}_i) = g'_t(\widetilde{z}_{i,t})$; otherwise i can be served a larger amount at time-slot t by the assumption that the derivatives are continuous. Therefore, we again have $\widehat{\lambda}_i = g'_t(\widetilde{z}_{i,t})$. □

Our objective is to establish an upper bound of $G(\widehat{\lambda}, \widehat{\mu})$ in terms of the objective value obtained by the online algorithm, namely, $\sum_i f_i(\overline{y}_i) - \sum_t g_t(\overline{z}_t)$. Our analysis is built upon the framework in [12]. We first consider the last term in (6), $\max_{x\in\mathcal{X}} \sum_t \left[\sum_i (\widehat{\lambda}_i - \widehat{\mu}_{it}) x_{it} - g_t(\sum_i x_{it})\right]$. Let $\{\widehat{x}_{it}\}$ denote the values of $\{x_{it}\}$ that maximize $\sum_i (\widehat{\lambda}_i - \widehat{\mu}_{it}) x_{it} - g_t(\sum_i x_{it})$ subject to the constraints (3) and (4). Let $j = \text{argmax}_{i:a_i\leq t\leq b_i} \widehat{\lambda}_i - \widehat{\mu}_{it}$. It is then clear that, without loss of optimality, we can set $\widehat{x}_{it} = 0$ for $i \neq j$, and the problem can be simplified to $\max_{x_{jt}\geq 0} (\widehat{\lambda}_j - \widehat{\mu}_{jt}) x_{jt} - g_t(x_{jt})$. Since $\widehat{\lambda}_i - \widehat{\mu}_{it} = g'_t(\widetilde{z}_{i,t})$ for any agent i with $\widetilde{x}_{it} > 0$, and the load of a time-slot never decreases, we can take j as any agent that is served in time-slot t in its final allocation, and the objective is maximized at $\widehat{x}_{jt} = \overline{z}_t$. Therefore, the last term in (6) becomes $\sum_t (g'_t(\overline{z}_t)\overline{z}_t - g_t(\overline{z}_t))$.

We then consider the first term in (6), $\max_{y\in\mathcal{Y}} \sum_i (f_i(y_i) - \widehat{\lambda}_i y_i)$. For each i, let $\widehat{y}_i$ denote the value of $y_i \leq Y_i$ that maximizes $f_i(y_i) - \widehat{\lambda}_i y_i$. Then we must have $f'_i(\widehat{y}_i) = \widehat{\lambda}_i$ or $\widehat{y}_i = Y_i$. By the definition of $\widehat{\lambda}_i$, we observe that $\widehat{y}_i = \widetilde{y}_i$. Hence, the first term becomes $\sum_i (f_i(\widetilde{y}_i) - \widehat{\lambda}_i \widetilde{y}_i)$. We then have

$$\begin{aligned} G(\widehat{\lambda}, \widehat{\mu}) &= \sum_i (f_i(\widetilde{y}_i) - \widehat{\lambda}_i \widetilde{y}_i) + \sum_{i,t} \widehat{\mu}_{it} X_i \\ &\quad + \sum_t (g'_t(\overline{z}_t)\overline{z}_t - g_t(\overline{z}_t)) \\ &= \sum_i f_i(\widetilde{y}_i) - \sum_i (\widehat{\lambda}_i \widetilde{y}_i - \sum_t \widehat{\mu}_{it} X_i) \\ &\quad + \sum_t (g'_t(\overline{z}_t)\overline{z}_t - g_t(\overline{z}_t)). \end{aligned} \tag{7}$$

Consider the second term in (7), we have

$$\begin{aligned} \sum_i (\widehat{\lambda}_i \widetilde{y}_i - \sum_t \widehat{\mu}_{it} X_i) &= \sum_i (\sum_t \widehat{\lambda}_i \widetilde{x}_{it} - \sum_t \widehat{\mu}_{it} X_i) \\ &\overset{(a)}{=} \sum_i \sum_t (\widehat{\lambda}_i - \widehat{\mu}_{it}) \widetilde{x}_{it} \\ &= \sum_i \sum_t g'_t(\widetilde{z}_{i,t}) \widetilde{x}_{it} \\ &\overset{(b)}{\geq} \sum_t \sum_i g'_t(\sum_{j\leq i} \overline{x}_{jt}) \overline{x}_{it} \\ &\overset{(c)}{\geq} \sum_t g_t(\overline{z}_t). \end{aligned} \tag{8}$$

where (a) follows from Lemma 4.1, (b) follows from the fact that x_{it} never increases after initial allocation in OAR and the convexity of g_t (recall that agents are sorted by their arrival times), and (c) follows from the convexity of g_t (see Figure 3 for an explanation).

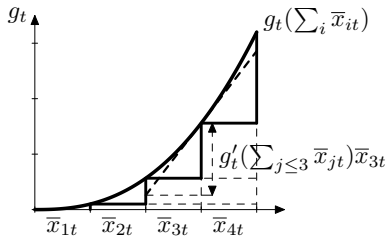


Figure 3: An example that shows $\sum_i g'_t(\sum_{j\leq i} \overline{x}_{jt})\overline{x}_{it} \geq g_t(\sum_i \overline{x}_{it})$ in (8).

From (7) and (8), we now have

$$G(\widehat{\lambda}, \widehat{\mu}) \leq \sum_i f_i(\widetilde{y}_i) + \sum_t g'_t(\overline{z}_t)\overline{z}_t - 2\sum_t g_t(\overline{z}_t). \tag{9}$$

Recall that our objective is to derive an upper bound of $G(\widehat{\lambda}, \widehat{\mu})$ in terms of $\sum_i f_i(\overline{y}_i) - \sum_t g_t(\overline{z}_t)$. To this end, we first make the following key observation, which establishes an upper bound for $\sum_i f_i(\widetilde{y}_i)$ in terms of $\sum_i f_i(\overline{y}_i)$ as shown in the lemma.

LEMMA 4.2. $\sum_i f_i(\widetilde{y}_i) \leq \frac{\gamma}{\gamma-1} \sum_i f_i(\overline{y}_i)$.

PROOF. We will show $\sum_i (\gamma-1)(f_i(\widetilde{y}_i) - f_i(\overline{y}_i)) \leq \sum_i f_i(\overline{y}_i)$, which implies the lemma. To simplify the description, we prove the statement for the discrete version of OAR. To this end, we view the charging opportunity in each time-slot as multiple units of size δ. To abuse the notation a little bit,

we redefine f_i in terms of units, and let $\widetilde{y}_i$ and $\overline{y}_i$ denote the number of units allocated to agent i in the beginning and in the end, respectively. The k-th unit that agent i obtains has a marginal valuation $v_{i,k} = f_i(k) - f_i(k-1)$. Assume $\overline{y}_i < \widetilde{y}_i$. User i went through a sequence of revocations after its initial allocation, where units of higher indices (and hence lower marginal valuations) are displaced first. All the units with index $\overline{y}_i$ and less are not reallocated. We note that by allowing units of different sizes, the following proof applies to the continuous case as well.

For agent i, each unit with index $\overline{y}_i + 1$ or higher went through a sequence of reallocations, represented by (i_1, k_1), (i_2, k_2), ..., (i_n, k_n), where in each pair, the first element denotes the agent that holds the unit, and the second element denotes the index of the unit for that agent. Each agent obtains the unit from the previous owner, and i_1 is the first agent that is allocated the unit and i_n the last. Let S denote the set of units over all the time-slots that has even been reallocated. For each unit $s \in S$, let $(i_1^s, k_1^s), (i_2^s, k_2^s), ..., (i_{n_s}^s, k_{n_s}^s)$ denote the corresponding sequence of reallocations. Define $V_s = \sum_{m=1}^{n_s-1} v_{i_m^s, k_m^s}$, the sum of marginal valuations over all the agents in the sequence except the last one. We then observe that $\sum_i (f_i(\widetilde{y}_i) - f_i(\overline{y}_i)) = \sum_{s \in S} V_s$, since any reallocation is with respect to a unit in S. On the other hand, $\sum_i f_i(\overline{y}_i) \geq \sum_{s \in S} v_{i_{n_s}^s, k_{n_s}^s}$, where the righthand side is the sum of marginal valuations of all the units in S with respect to the final agent that has the unit. The inequality follows from the fact that some units, once allocated to an agent, are never reallocated.

By the above observation, to prove the lemma, it is then sufficient to show that $(\gamma - 1)V_s \leq v_{i_{n_s}^s, k_{n_s}^s}$ for any s. Consider one such unit with sequence $(i_1, k_1), (i_2, k_2), ..., (i_n, k_n)$, where s is omitted to simplify the notation. According to OAR, (i_{m-1}, k_{m-1}) is displaced by (i_m, k_m) only if $v_{i_m, k_m} \geq \gamma v_{i_{m-1}, k_{m-1}}$. We prove by induction on $n \geq 2$. For $n = 2$, $(\gamma - 1)V = (\gamma - 1)v_{i_1,k_1} \leq v_{i_2,k_2}$. Assume the statement holds for $n \leq r$. For $n = r + 1$, we have

$$\begin{aligned}(\gamma-1)V =& (\gamma-1)\sum_{m=1}^{r} v_{i_m,k_m}\\ =& (\gamma-1)\sum_{m=1}^{r-1} v_{i_m,k_m} + (\gamma-1)v_{i_r,k_r}\\ \leq& v_{i_r,k_r} + (\gamma-1)v_{i_r,k_r}\\ =& \gamma v_{i_r,k_r}\\ \leq& v_{i_{r+1},k_{r+1}}.\end{aligned}$$

□

We then establish connections between the total valuation obtained and the total cost incurred by our algorithm in the following lemma and its corollary, which paves the way toward our main result.

LEMMA 4.3. $\gamma \sum_i f_i'(\overline{y}_i)\overline{y}_i \geq \sum_t g_t'(\overline{z}_t)\overline{z}_t$.

PROOF. Consider any time instance in serving requests using algorithm OAR. Let x_{it} denote the current allocation made for agent i in time-slot t, y_i the total allocation currently made for agent i, and z_t the current load in time-slot t. We claim that $\gamma f_i'(y_i) \geq g_t'(z_t)$ for any t such that $x_{it} > 0$. Otherwise, there must be a piece of demand from another agent j with derivative at least $\gamma f_i'(y_i)$ that is served by increasing the load at t, which, however, should have been served by displacing the allocation of i. It follows that $\gamma f_i'(\overline{y}_i) \geq g_t'(\overline{z}_t)$ for any t where $\overline{x}_{it} > 0$. Therefore,

$$\begin{aligned}\gamma \sum_i f_i'(\overline{y}_i)\overline{y}_i &= \sum_i \sum_t \gamma f_i'(\overline{y}_i)\overline{x}_{it}\\ &\geq \sum_i \sum_t g_t'(\overline{z}_t)\overline{x}_{it}\\ &= \sum_t g_t'(\overline{z}_t) \sum_i \overline{x}_{it}\\ &= \sum_t g_t'(\overline{z}_t)\overline{z}_t.\end{aligned}$$

□

We further have the following corollary.

COROLLARY 4.1. $\sum_i f_i(\overline{y}_i) \geq \frac{\Phi_g}{\gamma \Phi_f} \sum_t g_t(\overline{z}_t)$.

PROOF. By the definition of Φ_f and Φ_g, we have

$$\begin{aligned}\sum_i f_i(\overline{y}_i) &\geq \sum_i f_i'(\overline{y}_i)\overline{y}_i/\Phi_f\\ &\geq \sum_t g_t'(\overline{z}_t)\overline{z}_t/(\gamma\Phi_f) \quad \text{(Lemma 4.3)}\\ &\geq \frac{\Phi_g}{\gamma\Phi_f}\sum_t g_t(\overline{z}_t).\end{aligned}$$

□

From (9), Lemmas 4.2 and 4.3, and Corollary 4.1, we have

$$\begin{aligned}G(\widehat{\lambda}, \widehat{\mu}) \leq& \sum_i f_i(\widetilde{y}_i) + \sum_t g_t'(\overline{z}_t)\overline{z}_t - 2\sum_t g_t(\overline{z}_t)\\ \overset{(a)}{\leq}& \frac{\gamma}{\gamma-1}\sum_i f_i(\overline{y}_i) + \sum_t g_t'(\overline{z}_t)\overline{z}_t - 2\sum_t g_t(\overline{z}_t)\\ \overset{(b)}{\leq}& \frac{\gamma}{\gamma-1}\sum_i f_i(\overline{y}_i) + \gamma\sum_i f_i'(\overline{y}_i)\overline{y}_i - 2\sum_t g_t(\overline{z}_t)\\ \overset{(c)}{\leq}& (\frac{\gamma}{\gamma-1} + \gamma\Phi_f)\sum_i f_i(\overline{y}_i) - 2\sum_t g_t(\overline{z}_t) \qquad (10)\\ \overset{(d)}{\leq}& \frac{(\frac{1}{\gamma-1} + \Phi_f)\Phi_g - 2\Phi_f}{\frac{1}{\gamma}\Phi_g - \Phi_f}\Big[\sum_i f_i(\overline{y}_i) - \sum_t g_t(\overline{z}_t)\Big]. \qquad (11)\end{aligned}$$

where (a) follows from Lemma 4.2, (b) follows from Lemma 4.3, (c) follows from the definition of Φ_f, and (d) follows from Corollary 4.1 and simple algebra. Given Φ_f and Φ_g, the coefficient in (11) can be minimized by choosing the revocation coefficient to be

$$\gamma^* = \frac{\Phi_g - 2 + \sqrt{\frac{\Phi_g^2}{\Phi_f} - (1 + \frac{1}{\Phi_f})\Phi_g + 2}}{\Phi_g - 1}. \qquad (12)$$

Therefore, we obtain the following main result:

THEOREM 4.1. *OAR is* $\frac{(\frac{1}{\gamma^*-1} + \Phi_f)\Phi_g - 2\Phi_f}{\frac{1}{\gamma^*}\Phi_g - \Phi_f}$*-competitive, where* $\gamma^* = \frac{\Phi_g - 2 + \sqrt{\frac{\Phi_g^2}{\Phi_f} - (1+\frac{1}{\Phi_f})\Phi_g + 2}}{\Phi_g - 1}$.

Remark 3: The fact that the total valuation obtained is at least a factor $\rho \triangleq \frac{\Phi_g}{\gamma\Phi_f}$ of the total cost incurred in algorithm OAR, as proved in Corollary 4.1, is critical for deriving the competitive factor in (11) from the weaker form of (10). It can be seen that for a fixed γ, a larger ρ implies a smaller

competitive factor. In particular, for a given problem instance, the factor ρ can be improved by replacing Φ_f and Φ_g with $\max_i \phi_{f_i}(\overline{y}_i)$ and $\min_t \phi_{g_t}(\overline{z}_t)$, respectively. A further improvement is discussed in Example 4 below.

4.3 Examples

We now apply Theorem 4.1 to some concrete examples of valuation and cost functions. In the examples, all the agents are assumed to have the same type of valuation functions and the system operator has the same type of cost functions in all the time-slots. But the parameters of the functions vary over agents and time, respectively.

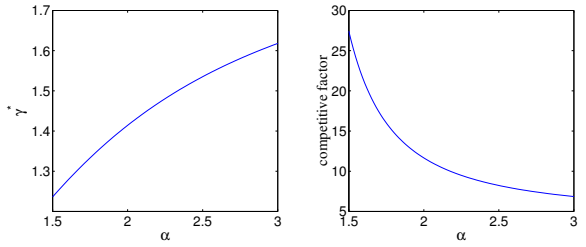
Example 1: $f_i(y_i) = v_i \log(1+y_i)$ for some $v_i > 0$, and $g_t(z_t) = c_t z_t^\alpha$ for some $c_t > 0$ and $\alpha \geq 1$. In this case, we have $\Phi_f = 1$ since $\phi_{f_i}(y_i) = \frac{[\log'(1+y_i)]y_i}{\log(1+y_i)} \to 1$ as $y_i \to 0$, and $\Phi_g = \alpha$ since $(z^\alpha)'z = \alpha z^\alpha$. Therefore, $\gamma^* = \frac{\alpha-2+\sqrt{(\alpha-1)^2+1}}{\alpha-1}$. In particular, consider the case when the cost function has a linear derivative, that is, $\alpha = 2$, we have $\gamma^* = \sqrt{2}$, and OAR is $\frac{2}{(\sqrt{2}-1)^2}(< 12)$-competitive. In general, the competitive factor obtained at γ^* increases as α approaches to 1 (see Figure 4(a)). On the other hand, we observe that $\phi_{f_i}(y_i)$ is decreasing on y_i and approaches to 0 as $y_i \to \infty$. Hence, for a given problem instance, when $\overline{y}_i$ is relatively large for most requests, a smaller competitive factor can be expected.

Example 2: $f_i(y_i) = v_i y_i^\beta$ for some $v_i > 0$ and $\beta \in [0,1]$, and $g_t(z_t) = c_t z_t^\alpha$ for some $c_t > 0$ and $\alpha \geq 1$. In this case, we have $\Phi_f = \beta$ and $\Phi_g = \alpha$. In particular, when $\alpha = 2, \beta = 1/2$, we have $\gamma^* = 2$, and OAR is 4-competitive. In general, the competitive factor increases as α/β approaches to 1 (see Figure 4(b)). However, we note that in the extreme case when $\alpha = \beta = 1$, that is, when both the valuation and the cost functions are linear, the algorithm OA proposed in Section 3.1 is optimal and revocation is not needed.

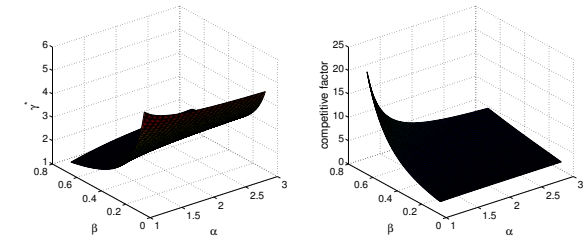
In both examples, we observe that the value of γ^* increases as α/β increases in most cases, and is minimized when $\alpha/\beta \to 1$. This can be explained from Corollary 4.1 and Lemma 4.2. When Φ_g/Φ_f is small, a small γ is needed to maintain the multiplicative factor in Corollary 4.1. On the other hand, when Φ_g/Φ_f is large, a large γ is desirable as it minimizes $\frac{\gamma}{\gamma-1}$, the multiplicative factor connecting the valuations of the initial and final allocation proved in Lemma 4.2.

Example 3 (free renewable energy): $g_t(z_t) = c_t[(z_t - z_t^0)^+]^\alpha$ for $c_t > 0, \alpha > 1, z_t^0 > 0$, where $(x)^+ \triangleq \max(x, 0)$. We use z_t^0 to model the amount of free renewable energy available at time t. Note that $g_t(z_t)$ is continuously differentiable for $\alpha > 1$, and $g_t'(z_t) = c_t\alpha[(z_t - z_t^0)^+]^{\alpha-1}$. Hence, $g_t'(z_t)z_t \geq \alpha g_t(z_t)$, and $\phi_{g_t} \geq \alpha$. Therefore, a non-zero renewable energy supply actually helps with the competitive performance (assuming it is predictable).

Example 4 (non-zero base load): $g_t(z_t) = c_t[(z_t+z_t^0)^2 - (z_t^0)^2]$ for some $c_t > 0, z_t^0 > 0$, where z_t^0 models the base load in the system that is out of the control of the operator. We have $\phi_{g_t} = 1$ since $\phi_{g_t}(z_t) = \frac{g_t'(z_t)z_t}{g_t(z_t)} = \frac{2(z_t+z_t^0)}{z_t+2z_t^0} \to 1$ as $z_t \to 0$. On the other hand, $\phi_{g_t}(z_t)$ is increasing on z_t and approaches 2 as $z_t \to \infty$. Therefore, a single worst-case ϕ_{g_t} is not very expressive. Below we outline an approach for improving Corollary 4.1, which can also be applied to other cost functions with increasing $\phi_{g_t}(z_t)$.



(a) Logarithmic valuation



(b) Power function valuation

Figure 4: Competitive performance for two types of valuation functions. In both (a) and (b), α is the power of the cost function, and in (b), β is the power of the valuation function.

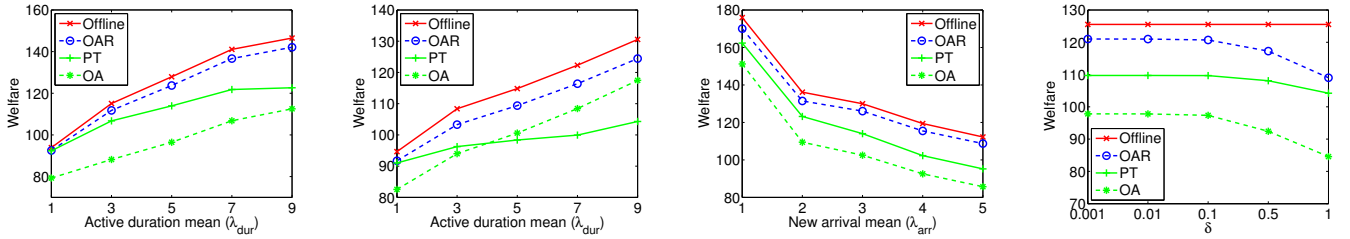
First, we define $\gamma_t = \max_{i:\overline{x}_{it}>0} \frac{g_t'(\overline{z}_t)}{f_i'(\overline{y}_i)}$. Then by a similar argument as in the proof of Lemma 4.3, we have $\gamma_t \leq \gamma$, and $\sum_i f_i'(\overline{y}_i)\overline{y}_i \geq \sum_t \frac{1}{\gamma_t} g_t'(\overline{z}_t)\overline{z}_t$. Now apply a similar proof of Corollary 4.1, we get $\sum_i f_i(\overline{y}_i) \geq \frac{1}{\Phi_f}\sum_t \frac{\phi_{g_t}(\overline{z}_t)}{\gamma_t} g_t(\overline{z}_t)$. For any time-slot with $\overline{x}_{it} > 0$, we must have $f_i'(\overline{y}_i) \geq g_t'(0)$. Therefore, $\gamma_t \leq \frac{g_t'(\overline{z}_t)}{g_t'(0)} = \frac{2(z_t+z_t^0)}{2z_t^0}$; hence, $z_t \geq (\gamma_t - 1)z_0$. It follows that $\phi_{g_t}(\overline{z}_t) \geq \frac{2((\gamma_t-1)z_0+z_0)}{(\gamma_t-1)z_0+2z_0} = \frac{2\gamma_t}{\gamma_t+1}$; hence, $\phi_{g_t}(\overline{z}_t)/\gamma_t \geq \frac{2}{\gamma_t+1} \geq \frac{2}{\gamma+1}$. Therefore, $\sum_i f_i(\overline{y}_i) \geq \frac{2}{(\gamma+1)\Phi_f}\sum_t g_t(\overline{z}_t)$. In contrast, if Corollary 4.1 is applied with $\Phi_g = 1$, we get $\sum_i f_i(\overline{y}_i) \geq \frac{1}{\gamma\Phi_f}\sum_t g_t(\overline{z}_t)$. Since $\frac{2}{(\gamma+1)} > \frac{1}{\gamma}$ whenever $\gamma > 1$, a smaller competitive factor can be obtained using this approach.

5. EVALUATION

In this section, we evaluate the performance of our online algorithm using simulations. We compare our OAR algorithm with PT and OA. The Simple-Increment approach discussed in Section 3.2 is used to implement all the three algorithms, where a discrete increment δ (i.e., the charging unit) is applied in each iteration. For our algorithm, the revocation coefficient γ is determined by Theorem 4.1. We study the performance of these algorithms under different values of δ, and compare them with the optimal offline solution (with continuous charging rate), obtained using the CVX toolbox [11]. Our results show that the OAR algorithm performs clearly better than PT and OA and achieves close to the offline optimal welfare under various settings. The simulation results also illustrate scenarios when PT or OA does not perform well.

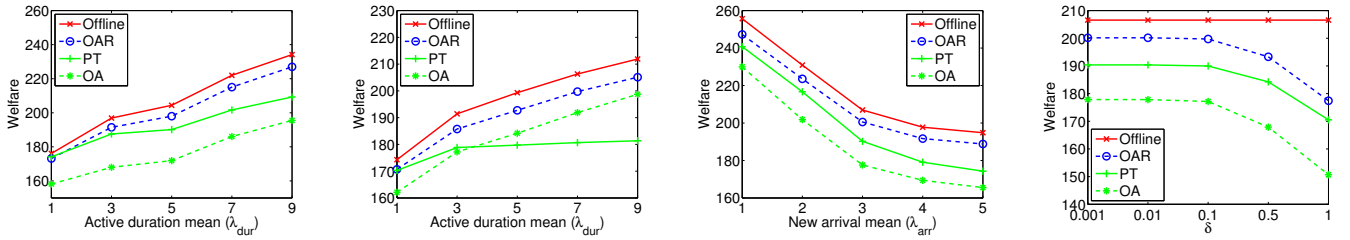
5.1 Setup

In our simulations, we assume that the number of new arrivals of charging requests in each time-slot follows a Poisson distribution with mean λ_{arr}, independent of other time-slots. The active duration of each request follows an ex-



(a) $\delta = 0.01, Y_i = 5, \lambda_{arr} = 3$ (b) $\delta = 0.01, Y_i = 1, \lambda_{arr} = 3$ (c) $\delta = 0.01, Y_i = 3, \lambda_{dur} = 5$ (d) $Y_i = 3, \lambda_{arr} = 3, \lambda_{dur} = 5$

Figure 5: Simulation results for logarithmic valuation function and quadratic cost.



(a) $\delta = 0.01, Y_i = 5, \lambda_{arr} = 3$ (b) $\delta = 0.01, Y_i = 1, \lambda_{arr} = 3$ (c) $\delta = 0.01, Y_i = 3, \lambda_{dur} = 5$ (d) $Y_i = 3, \lambda_{arr} = 3, \lambda_{dur} = 5$

Figure 6: Simulation results for power valuation function and quadratic cost.

ponential distribution of mean λ_{dur}, independent of other requests. All the time-slots have the same cost function $g_t(z_t) = z_t^2$. All the requests have the same type of valuation functions. Two types are considered: $f_i^1(y_i) = v_i \log(1 + y_i)$ and $f_i^2(y_i) = v_i\sqrt{y_i}$. The coefficients v_i are generated from a uniform distribution in [1,10]. All the requests have the same charging rate limit $X_i = 1$ and the same maximum charging amount Y_i. The charging unit δ is chosen from $\{0.001, 0.01, 0.1, 0.5, 1\}$. Each figure below illustrates the average results over 50 independent scenarios generated under a given set of parameters, where 50 requests are generated in each scenario.

5.2 Results

The simulation results for logarithmic valuation function together with the quadratic cost are given in Figure 5. We first set $\delta = 0.01$ and plot the results in Figures 5(a)-5(c). Since δ is small enough compared with X_i and Y_i, the results closely reflect what can be expected in the continuous charging rate regime. In Figure 5(a), we fix $Y_i = 5$ and $\lambda_{arr} = 3$, and plot the welfare achieved by different algorithms versus the active duration mean (λ_{dur}). We make the following observations: (1) Our algorithm achieves close to optimal welfare; (2) All the algorithms achieve better welfare for a larger λ_{dur}, which reflects the benefit of introducing demand-side time elasticity; (3) The gap between PT and our algorithm (and the offline optimal) increases for larger λ_{dur}, which is due to the fact that PT does not take the time elasticity of requests into account explicitly when making charging decisions; and (4) Compared with PT, OA can better utilize large λ, but it has the lowest welfare among the four algorithms. The situation changes when we fix $Y_i = 1$ instead as shown in Figure 5(b). In this case, OA performs better than PT except when λ_{dur} is very small. The intuition is that a small Y_i implies that the traffic load is relatively low (compared to the cost). Hence, there is an opportunity to flatten the load to reduce the cost when time elasticity allows, which, however, is not well utilized by PT. From Figures 5(a) and 5(b), we also observe that our algorithm achieves close to optimal performance even when the average time elasticity is low, while the performance of PT and OA vary under different scenarios.

Figure 5(c) shows the impact of mean arrival rate (λ_{arr}) on the performance of the algorithms. Since we fix the number of requests to be 50, a larger λ_{arr} leads to a lower welfare due to the higher density of the load. Moreover, since the number of new requests in each time-slot follows a Poisson distribution, a larger λ_{arr} also leads to a larger variance in the arrival process. We observe that our algorithm always achieves close to optimal performance. On the other hand, the gap between PT and the optimal increases for a larger λ_{arr}. This is also the case for OA, and can be more easily seen when the valuation functions are power functions (see Figure 6(c)). For PT, the problem is due to the fact that time elasticity has been largely ignored, which, however, is beneficial especially when the variance in workload is high. On the other hand, OA suffers from the weakness that requests of high valuations can be blocked especially when the system load is high.

We then study the impact of different charging unit δ. The results are given in Figure 5(d), where we fix $\lambda_{arr} = 3, \lambda_{dur} = 5$, and vary δ in $\{0.001, 0.01, 0.1, 0.5, 1\}$. The offline optimal is still computed by assuming a continuous charging rate since finding an optimal solution for a discrete charging rate is computationally difficult. We observe that there is little performance loss for the three online algorithms when δ is changed from 0.001 to 0.1. Therefore, a good competitive performance can be achieved at a relatively low complexity (recall that OAR has a time complexity of $O(\frac{Y}{\delta}N^2D)$ as discussed in Section 3.2). On the other hand, the performance of the online algorithms clearly drop when δ is lose to 1, the charging rate limit, due to the inherent integrality gap. However, we observe that our algorithm still performs better than the other two even in this regime.

The simulation results for power valuation function and quadratic cost are given in Figure 6, where we observe sim-

ilar trends as in the logarithmic valuation case discussed above.

6. CONCLUSION

To improve the energy efficiency in supporting large scale EV charging, an effective approach is to study coordinated charging schemes that can exploit the flexibilities at both the demand side and the supply side. In this work, we propose an online algorithm for scheduling deferrable charging requests to balance the total value of vehicle owners and the total cost for providing charging service. Assuming that the charging rate is continuous, we characterize the competitive performance of our algorithm in terms of the concavity of the valuation function and the convexity of the cost function. Numerical results demonstrate that our algorithm achieves close to optimal performance even for discrete charging rates.

7. ACKNOWLEDGMENTS

This work is supported in part by a grant from the National Science Foundation ECCS-1232118.

8. REFERENCES

[1] A. Subramanian, M. Garcia, A. Domínguez-García, D. Callaway, K. Poolla, and P. Varaiya. Real-time scheduling of deferrable electric loads. In *Proc. of ACC*, 2012.

[2] D. P. Bertsekas. *Nonlinear Programming, 2nd edition*. Athena Scientific, 1999.

[3] A. Blum, A. Gupta, Y. Mansour, and A. Sharma. Welfare and profit maximization with production costs. In *Proc. of FOCS*, 2011.

[4] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 2005.

[5] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing. In *Proc. of the 13th Annual European Symposium on Algorithms*, 2005.

[6] N. Buchbinder and J. Naor. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science*, 3(2-3):93–263, 2007.

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

[8] S. Chen and L. Tong. iEMS for large scale charging of electric vehicles: Architecture and optimal online scheduling. In *Proc. of SmartGridComm*, 2012.

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

[10] E. H. Gerding, V. Robu, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings. Online mechanism design for electric vehicle charging. In *Proc. of AAMAS*, 2011.

[11] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`, Sept. 2013.

[12] A. Gupta, R. Krishnaswamy, and K. Pruhs. Online primal-dual for non-linear optimization with applications to speed scaling. In *10th Workshop on Approximation and Online Algorithms (WAOA)*, 2012.

[13] T. Koslowski. The electric vehicle's value chain and technology evolution. Technical report, Gartner Inc., 2009.

[14] I. Koutsopoulos and L. Tassiulas. Optimal control policies for power demand scheduling in the smart grid. *IEEE Journal on Selected Areas in Communications*, 30(6):1049–1060, 2012.

[15] T. Lan, D. Kao, M. Chiang, and A. Sabharwal. An axiomatic theory of fairness in network resource allocation. In *Proc. of IEEE Infocom*, 2010.

[16] N. Li, L. Chen, and S. H. Low. Optimal demand response based on utility maximization in power networks. In *Power and Energy Society General Meeting*, 2011.

[17] B. Obama and J. Biden. New Energy for America. `http://energy.gov/sites/prod/files/edg/media/Obama_New_Energy_0804.pdf`, 2009.

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

[19] V. Robu, S. Stein, E. H. Gerding, D. C. Parkes, A. Rogers, and N. R. Jennings. An online mechanism for multi-speed electric vehicle charging. In *Proc. of AMMA*, 2011.

[20] S. Stein, E. H. Gerding, V. Robu, and N. R. Jennings. A model-based online mechanism with pre-commitment and its application to electric vehicle charging. In *Proc. of AAMAS*, 2012.

[21] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced cpu energy. In *Proc. of FOCS*, 1995.