Competitive Online Scheduling Algorithms with Applications in Deadline-Constrained EV Charging

Bahram Alinia*, Mohammad Sadegh Talebi†, Mohammad H. Hajiesmaili‡,
Ali Yekkehkhany§, and Noel Crespi*
* Institut Mines-Telecom, Telecom SudParis, 91000 Evry, France
† Department of Automatic Control, EECS, KTH Royal Institute of Technology, Stockholm, Sweden
‡Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, USA
§University of Illinois at Urbana-Champaign, IL, USA
Emails: bahram.alinia@telecom-sudparis.eu, mstms@kth.se, hajiesmaili@jhu.edu,
yekkehk2@illinois.edu, noel.crespi@mines-telecom.fr

***Abstract*—This paper studies the classical problem of online scheduling of deadline-sensitive jobs with partial values and investigates its extension to Electric Vehicle (EV) charging scheduling by taking into account the processing rate limit of jobs and charging station capacity constraint. The problem lies in the category of time-coupled online scheduling problems without availability of future information. This paper proposes two online algorithms, both of which are shown to be $(2-\frac{1}{U})$-competitive, where U is the maximum scarcity level, a parameter that indicates demand-to-supply ratio. The first proposed algorithm is deterministic, whereas the second is randomized and enjoys a lower computational complexity. When U grows large, the performance of both algorithms approaches that of the state-of-the-art for the case where there is processing rate limits on the jobs. Nonetheless in realistic cases, where U is typically small, the proposed algorithms enjoy a much lower competitive ratio. To carry out the competitive analysis of our algorithms, we present a proof technique, which is novel to the best of our knowledge. This technique could also be used to simplify the competitive analysis of some existing algorithms, and thus could be of independent interest.**

I. Introduction

Online scheduling of heterogeneous deadline-constrained jobs in the presence of limited resources is a fundamental, yet challenging problem in various application scenarios. Notable examples are network buffer management [1], [2], processors sharing [3], [4], as traditional applications, and cloud job scheduling [5] and electric vehicle (EV) charging scheduling [6]–[8], as the state-of-the-art examples.

In the classic form of the online scheduling of deadline-sensitive jobs, there is a limited resource (e.g., router's buffer, CPU time, or the maximum power capacity of EV charging station) that is shared among a set of jobs (users, task, or EVs) that arrive over time in an online fashion. The jobs are heterogeneous in terms of arrival, deadline, demand, and value (or weight), and the goal is to maximize total value obtained from the jobs, subject to the resource capacity constraints. The target applications could be categorized into *full* [9], [10] and *partial* [11]–[17] execution models. In the present work, we focus on the latter, where partially completed jobs get partial values proportional to their received resource. Notable examples of partial models are job scheduling in web search applications [18], multimedia content transmission [11], and EV charging scheduling [8].

The underlying classic problem under partial execution model has been first introduced in [11], where two simple greedy heuristics are proposed. Our focus in this paper is on online algorithms with a bounded worst-case performance determined by their *competitive ratio*[1] to maximize charging station gain. Using the competitive analysis [19], the authors in [11] demonstrated that both algorithms achieve a competitive ratio of 2. However, the problem setup in [11] does not take into account the maximum processing rate of the jobs. With extensive applications in the recent research topics, the problem has been extended to several other settings such as multi-resource allocation [17], providing resource commitment [20], and truthful analysis [10], among others. We review the related literature on these in Section II.

This paper especially focuses on the application of scheduling that is identified with the advent of EVs. EVs are a promising alternative for the conventional vehicles considering their significant advantages in energy efficiency, zero emission, and relieve reliance on fossil fuels. With increasing number of EVs, their charging demand can pose a tremendous challenge to the power system operation [6]–[8]. EV charging demand, however, is usually deferrable implying that there is often considerable flexibility in charging schedule.

It turns out that the problem of EV charging scheduling in a charging station and the cloud job scheduling problem share some similarity in structure. Similarly to the job scheduling problem, EVs arrive to the charging station in an online fashion, each of which with different arrival time, deadline, demand, and value. The resource constraint in EV charging scenario is the limited power of the charging station to be allocated to the EVs at each time slot. The power constraint is determined by the chargers' or transformers' output power or is set manually by the station operator. Despite these similarities, EV scheduling problem poses an additional constraint that makes the corresponding classic job scheduling problem more

[1] An online algorithm $\mathcal{A}$ is c-competitive for $c \geq 1$ if for any input instance the optimal gain is at most c times the algorithm's gain.

challenging. More specifically, the input power of EV's battery is limited to a specific amount called *maximum charging rate*. Therefore, unlike the traditional scheduling problems, the completion time of a demand in EV scheduling problem not only depends on the availability of the resources, but it is also dependent on the maximum charging rate of its battery.

In this paper, we revisit the deadline constrained job scheduling problem with partial values and limiting maximum processing rate of the jobs, and make the following key contributions:

1) We propose a deterministic online algorithm, WFAIR, along with a simple randomized algorithm, WRAND, for the EV charging scheduling problem with capacity constraint of charging stations and maximum charging rate constraint. We show that both algorithms are $(2-\frac{1}{U})$-competitive, where U is the maximum scarcity level of the system (see Definition 1 for a formal definition). To the best of our knowledge, amongst existing algorithms capable of respecting processing limit of the jobs, none of them attains a competitive ratio better than 2.
2) We examine the performance of the proposed algorithms by trace-driven experiments. As our results show, the empirical cost ratios of our algorithms are much better than the obtained theoretical competitive ratios.
3) To accomplish the competitive analysis of the two algorithms, we propose a new proof technique that can be applied to a wider class of deadline-constrained online scheduling problems beyond this work. In particular, when applied to derive competitive performance bound of an existing algorithm in [11], the presented technique recovers the same results using a simpler proof. We therefore believe that it could be of independent interest beyond EV charge scheduling problem as well.

The rest of this paper is organized as follows. In Section II, we review the literature. In Section III, the tailored system model for EV charging application is introduced and the problem is formulated. Section IV proposes two deterministic and randomized algorithms. Section V is devoted to the competitive analysis of the proposed algorithms. Simulation results are reported in Section VI. Finally, Section VII concludes the paper and highlights future directions.

II. RELATED WORKS

The job scheduling problem has appeared in different application domains including task scheduling in processors [3], [4], cloud computing [5], [10], [16], [17], and network buffer management [1]. In this problem, a decision maker aims to maximize the total value of processed jobs in the presence of deadline and resource constraints under heterogeneity in the value of jobs. In this section, we first look over the existing works in the related domains (Section II-A), and then review the EV scheduling problem (Section II-B).

A. Classic Job Scheduling Algorithms

As there is a plethora of real-world applications for the problem, extensive studies have been conducted on the basic form of the problem [1], [10]–[18], [20]–[23] with a focus on online algorithm design. These studies can be classified into full execution and partial execution models. As our problem belongs to the partial execution category, we only review works related to partial execution. We refer to [9] as the offline result, and [1], [10], [20]–[23] as online results.

Studies in [11]–[18] considered partial execution model considered in this paper. Two simple and natural greedy algorithms named FIRSTFIT and ENDFIT are proposed in [11], where both algorithms are 2-competitive and the bound is tight. For non-decreasing concave utility functions, the ISPEED algorithm in [16] provides competitive ratio of $2+\alpha$, where α is a shape parameter. The study is extended in [17] to the case of multi-resource scenario by taking into account the processing limit of jobs and providing a competitive ratio of 2. In [14], the authors provide a lower bound of 1.236 for the competitive ratio and propose MIXED, which is shown to be 1.8-competitive. An improvement to this result appears in [13], where the authors propose MIX, and show that it is $\frac{e}{e-1} \approx 1.582$-competitive. The idea is that each job receives some resources according to its unit value unless its unit value is less than a threshold. Furthermore, a lower bound of 1.25 is provided for the competitive ratio of any randomized (and hence deterministic) online algorithm. We stress that filling the gap between the lower and the upper bounds is still an open problem. Moreover, authors provide an upper-bound of 1.618 when time sharing is not allowed (i.e., only one job can be processed at each time). [12] studied the problem when time sharing is not allowed and the number of concurrent jobs, m, is limited. Their proposed algorithm, GAP, is 1.618-competitive when there are only two concurrent jobs. However, GAP attains a larger competitive ratio when the number of concurrent jobs increases. Studies in [15], [18] address scheduling in interactive services such as web servers and finance services but do not provide competitive analysis for the proposed methods. We emphasize that in our model jobs (EVs' charging demand in our case) have limited processing rate, which adds to the complexity of the problem.

B. EV Charging Scheduling

There is a growing number of studies in the EV scheduling problem (see, e.g., [8], [24]) to provide efficient algorithms aiming to optimize different objective functions including aggregator profit, users' comfort level, etc. In this section, we will focus on the studies that propose competitive algorithms, i.e., those algorithms whose worst-case performance with respect to the optimal offline solution is bounded.

The EV scheduling problem is a special case of job scheduling problem where the processing limit of jobs is an essential constraint to be considered. Therefore, the studies reviewed in Section II cannot be directly applied to the EV problem. Although there are some exceptions [9], [17], [21], yet none of them provide a competitive ratio better than 2.

Moreover, [9], [21] consider a slackness parameter in their model (see Section II-A), which we believe cannot capture the real world scenarios. Also, the algorithm in [17] reduces to FIRSTFIT algorithm [11] which is compared to our algorithms in simulation section. An online $\frac{e}{e-1} \approx 1.582$-competitive algorithm is developed in [25] but the constraint on the charging speed of the EVs is missing from the formulation. Moreover, the authors assume that all EVs have the same demand. Assuming that there is no resource constraint in charging station and the objective is to minimize the cost for the aggregator, [26] and [27] proposed online algorithms, called SOCA and ORCHARD respectively, that achieve the optimal competitive ratio of 2.39. The studied problem in this paper is fundamentally different than [26] and [27] in the constraint set and the objective function. [28] considers the same model as described in this paper and proposes a truthful online scheduling algorithm assuming that discharging of EVs can be done instantaneously in their departure (referred to as on-departure burning) and EVs have the same charging rate. The authors extended the work in [29] and [30] to the case of heterogeneous charging rates, and proved that the proposed algorithm is 2-competitive. However, the assumption of on-departure burning is not realistic. [31] proposed the TAGS algorithm and proved that it attains an optimal competitive ratio. However, in their model all EVs have the same unit value and there is no limit on the charging speed. Under the same model, [7] proposed DSAC, an online scheduling algorithm with admission control (i.e., on-arrival notification). DSAC achieves an optimal competitive ratio for the case of linear utility function. In this paper, we propose competitive online algorithms for EV scheduling problem where their competitive ratios are less than the best known results under fairly reasonable assumptions. To the best of our knowledge, there is no online algorithm achieving a competitive ratio better than 2 that respects charging rate limitations of the EVs (or jobs).

III. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a discrete time-slotted system where the time horizon is divided into T time slots indexed by $t \in \mathcal{T} := \{1, \dots, T\}$. Time slots are assumed to be of equal lengths. We present our model in the context of EV charging scheduling. Consider a single charging station with capacity (resource) constraint of C (in kWh, say) to serve a set comprising n users (EVs, or jobs, used interchangeably) indexed by j. User j is represented by its demand profile $\pi_j = \langle \mathcal{T}_j, v_j, D_j \rangle$, where $\mathcal{T}_j$ denotes the *availability window* and v_j is the value for receiving demand D_j. The availability window $\mathcal{T}_j$ consists of all time slots from the arrival to the departure of user j. Let K denote the maximum charging rate of the EV, which is assumed to be fixed for all EVs[2]. We denote by $\rho_j = \frac{v_j}{D_j}$ the *unit value* (a.k.a. marginal value [9] or value density [22]) of user j.

[2]Our algorithms can be straightforwardly extended to the setting with heterogeneous charging rate demands. We consider fixed rates to facilitate our competitive analysis.

TABLE I: Summary of notations

Notation	Description
$\mathcal{T}$	Set of time slots with $\lvert\mathcal{T}\rvert = T$, indexed by t
$\mathcal{N}_t$	Set of available users at t with $\lvert\mathcal{N}_t\rvert = n_t$
$\mathcal{M}_t$	Set of active users at t with $\lvert\mathcal{M}_t\rvert = M_t$
$\mathcal{T}_j$	Availability window of user j
D_j	Demand of user j
v_j	Value of user j for receiving demand D_j
K	The maximum charging rate of users
ρ_j	Unit value of user j, i.e., $\rho_j = v_j/D_j$
$R_{j,t}$	Residual demand of user j at t: $D_j - \sum_{t' \in \mathcal{T}_j, t' \le t} y_{j,t'}$
C	Capacity constraint of the charging station
$y_{j,t}$	**decision variable**, the amount that user j is charged at t

We consider an online setting in which the profile of each user is only known to the scheduler upon its arrival.

We assume a preemptive model in which the scheduler is allowed to pause charging of an EV at any time and resume it later. We denote by $y_{j,t} \in [0, K]$ the allocated resource to the EV j at slot t. Moreover, $R_{j,t} = D_j - \sum_{t':t' \le t} y_{j,t'}$ is the residual demand of EV j at time slot t. We consider the partial charging model, where if EV j receives its total demand D_j within its availability window, the obtained value is v_j. Otherwise, its gain would be $\sum_{t \in \mathcal{T}_j} y_{j,t} \rho_j$.

Next we introduce some definitions. We say that EV j is *available* at time slot t if $t \in \mathcal{T}_j$. Moreover, given the scheduling policy, EV j is *active* at time slot t if it is available at t but its charging demand is not fulfilled yet. Finally, EV j is said to be *selected* at time slot t if $y_{j,t} > 0$. For any time t, let $\mathcal{N}_t$ and $\mathcal{M}_t$ denote the set of available and active EVs at time slot t, respectively. Further, let n_t and M_t be the cardinality of $\mathcal{N}_t$ and $\mathcal{M}_t$, respectively. Under a given algorithm $\mathcal{A}$, introduce $\mathcal{S}_{\mathcal{A},t}$ as the set of selected EVs at time t by $\mathcal{A}$:

$$\mathcal{S}_{\mathcal{A},t} := \{j : y^{\mathcal{A}}_{j,t} > 0\}.$$

Moreover, we define $\mathcal{S}_{\text{OPT},t} := \{j : y^{\star}_{j,t} > 0\}$, where $y^{\star}_{j,t}$ is the allocated resource to j at t by the optimal solution.

The key notations used in this paper are listed in Table I.

Having introduced these notations and definitions, we may formulate the EV scheduling problem under partial execution model as follows:

$$\text{RJSP}: \quad \max_{\vec{y}} \quad \sum_{j=1}^{n} \rho_j \sum_{t \in \mathcal{T}_j} y_{j,t} \tag{1a}$$

$$\text{s.t.} \quad \sum_{t \in \mathcal{T}_j} y_{j,t} \le D_j, \qquad \forall j \tag{1b}$$

$$\sum_{j : t \in \mathcal{T}_j} y_{j,t} \le C, \qquad \forall t \tag{1c}$$

$$0 \le y_{j,t} \le K, \quad \forall j, t, \tag{1d}$$

$$y_{j,t} = 0, \quad \forall (j,t) : t \notin \mathcal{T}_j \tag{1e}$$

The RJSP in Eq. (1a) maximizes the charging station gain. The constraint in (1b) limits the total resources received by an EV to its demand as there is no benefit for the charging

station to overcharge the EVs. The second constraint in (1c) is the capacity constraint, and the third and the fourth constraints enforce the charging station to respect the maximum charging rate and to charge EVs only during their availability window.

First observe that RJSP is a linear program and can hence be solved efficiently in offline scenarios. Second, in online scenarios the problem is less challenging to solve if the charging rate constraint in (1d) is omitted. In fact, with the charging rate constraint, part of the resources at some time slots might remain unused while there are some users that have not received their entire demand yet. Such users may also not receive their total demand in the next time slots if they are not selected for charging due to resource scarcity. Third, any c-competitive algorithm for RJSP is also a c-competitive solution for the basic form of RJSP (i.e., the form without maximum charging rate constraint). However, the inverse is not necessarily true.

IV. Online Scheduling Algorithms

In this section, we propose two online algorithms for RJSP. The competitive ratios of the two algorithms as well as their computational complexities are summarized in Table II.

TABLE II: Summary of the proposed algorithms

Alg.	comp. ratio	Complexity	Type
WFair	$2-\frac{1}{U}$	$O(n^2T)$	Deterministic
WRand	$2-\frac{1}{U}$	$O(nT\log n)$	Randomized

A. The WFair Algorithm

In this subsection, we present a deterministic algorithm, which we refer to as WFair, as an online algorithm for RJSP. The pseudo-code of WFair is listed as Algorithm 1.

WFair allocates the available resources to the users proportional to their unit values. More precisely, at each time slot t, the algorithm runs in multiple rounds, where at each round an active user j receives

$$\min\Big\{\frac{\rho_j}{\sum_{i\in\mathcal{M}_t}\rho_i}\Big(C-\sum_{i\in\mathcal{N}}y_{i,t}\Big), R_{j,t}, K-y_{j,t}\Big\} \qquad (2)$$

units of the resource (Line 6 of the algorithm). The received resource by each user is linearly correlated to its unit value. Therefore, for all active users at t, it holds that $y_{j,t}>0$ as unit values are non-zero, i.e., no user will be left unallocated but it may receive an infinitesimal amount if $\frac{\rho_j}{\sum_{i\in\mathcal{M}_t}\rho_i}$ is very small. Note that for some users, the second or the third term in Eq. (2) might be selected. In this case, the aggregate allocated amount might be less than the total capacity. This potential issue is resolved by re-allocating the residual resource in multiple iterations until the entire resource allocated or all the active users get their maximum possible requirements.

We stress that this allocation rule is in contrast to that of FirstFit algorithm [11], which allocates the resources to the most valuable users first.

Fig. 1 shows a general example, which we will refer to frequently to clarify the technical discussions. If we run WFair over the scenario of Fig. 1, it will share the resources in the first time slot equally between users 1 and 2 (because $\rho_1=\rho_2$) and set $y_{1,1}=y_{2,1}=0.5$. Therefore, the gain (i.e., total valuation of allocated resources) of WFair at $t=1$ is 1. The worst-case for WFair happens when no user arrives at $t=2$. In this case, Opt $=2$ (by allocating user 2 in the first slot and user 1 in the second slot) and WFair will set $y_{1,2}=0.5$. So, the total gain by WFair is 1.5.

Algorithm 1: WFair (for time slot t)

1 $\mathcal{L}_t \leftarrow \mathcal{M}_t$
2 $y_{j,t}\leftarrow 0,\ \forall j$
3 **while** $\sum_j y_{j,t}<C$ **and** $\mathcal{L}_t\neq\emptyset$ **do**
4 **for** *all* $j\in\mathcal{L}_t$ **do**
5 $\delta_{j,t}\leftarrow$ $\min\big\{\frac{\rho_j}{\sum_{i\in\mathcal{L}_t}\rho_i}\big(C-\sum_{i\in\mathcal{N}_t}y_{i,t}\big), R_{j,t}, K-y_{j,t}\big\}$
6 **for** *all* $j\in\mathcal{L}_t$ **do**
7 $y_{j,t}\leftarrow y_{j,t}+\delta_{j,t}$
8 $R_{j,t}\leftarrow R_{j,t}-\delta_{j,t}$
9 **if** $R_{j,t}=0$ **then**
10 $\mathcal{L}_t\leftarrow\mathcal{L}_t\backslash j$

We now illustrate a worst-case instance for WFair. For more in-depth analysis on the competitive ratio of WFair see Section V.

Worst-case instance for WFair: Consider a single time slot scenario with $C=K=T=1$ and n users where n is sufficiently big. The charging profiles are $\pi_1=\langle\{1\},\frac{1}{2},\frac{1}{2}\rangle$ and $\pi_2=\cdots=\pi_n=\langle\{1\},\frac{1}{2n},\frac{1}{2}\rangle$. Hence, we have $\rho_1=0.5$ and $\rho_2=\cdots=\rho_n=\frac{1}{2n}$. Therefore, $\sum_j\rho_j=0.5+\frac{n-1}{2n}$ which approximates to 1 as n is large. The optimal solution is to fully schedule user 1 while giving no resources to the other users. WFair shares the resources between all the EVs such that EV 1 only receives half of the resource and the other half is allocated to rest of the users. This leads to a total gain of 0.5 while the optimal gain is 1.

The above example indicates that the competitive ratio of WFair could not be better than 2. We however note that the presented worst-case scenario is quite unrealistic as the ratio of demand-to-supply is a small constant in practice. Under this assumption, the competitive ratio can be improved.

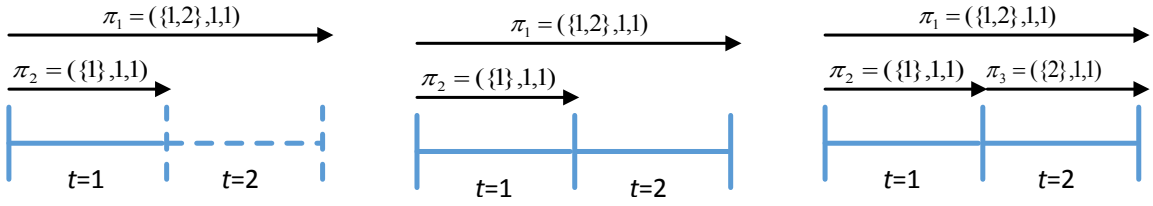
Next we define the notion of scarcity level:

Definition 1 (Scarcity Level [32])**.** *The* scarcity level U_t *at time slot* t *is defined as* $U_t=\frac{n_tK}{C}$*. Moreover, the maximum scarcity level of the system is* $U=\max_t U_t$*.*

Indeed the scarcity level U_t is an indication of *demand-to-supply ratio*, where the demand is roughly U_t times higher than the available resource.

The following theorem provides the competitive ratio of WFair:

Theorem 1. WFair *is* $(2-\frac{1}{U})$*-competitive.*



(a) Jobs 1 and 2 arrive at $t = 1$. (b) **Scenario 1:** No arrival at $t = 2$. (c) **Scenario 2:** Job 3 arrives at $t = 2$.

Fig. 1: A simple scheduling scenario. Dotted line indicates a time slot that is not visited yet and the scheduler has no information about the arriving EVs in that slot.

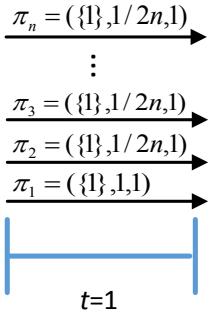


Fig. 2: A worst-case instance for WFAIR.

Algorithm 2: WRAND (for time slot t)

1 $\mathcal{L}_t \leftarrow \mathcal{M}_t$
2 $y_{j,t} \leftarrow 0, \forall j$
3 **while** $\sum_i y_{i,t} < C$ **and** $\mathcal{L}_t \neq \emptyset$ **do**
4 Select user j with probability $\frac{\rho_j}{\sum_{i\in\mathcal{L}_t}\rho_i}\mathbb{I}_{\{j\in\mathcal{L}_t\}}$
5 $y_{j,t} \leftarrow \min\{K, R_{j,t}, C - \sum_i y_{i,t}\}$
6 $\mathcal{L}_t \leftarrow \mathcal{L}_t \backslash j$

B. *The* WRAND *Algorithm*

In this subsection, we present WRAND, a *randomized* algorithm for RJSP. In general, randomized algorithms bring two main advantages over deterministic ones. First, they are usually more efficient in terms of the algorithm cost. Second, a randomized algorithm usually admits a simpler design than deterministic ones, which in turn makes the implementation easier. The competitive ratio of a randomized algorithm is measured with respect to an *adversary model*, which determines the way the input sequence to the problem is generated. We distinguish between two notions of adversary: *oblivious adversary* and *adaptive online adversary*. An oblivious adversary knows the algorithm code but should choose the entire input sequence in advance (i.e., before the start of the algorithm), whereas an adaptive online adversary, can well condition the input at each time step on the algorithm's history of plays.

The WRAND algorithm is motivated as follows (we refer to Algorithm 2 for its pseudo-code). At each slot t, the algorithm selects one or multiple active users randomly with a probability proportional to their unit values: the more the unit value of a user, the higher the probability it will be selected. More specifically, the algorithm maintains a set $\mathcal{L}_t$ that comprises all active users whose demand has not been met. Then, at each round of the 'while' loop, it selects a user j with probability proportional to $\rho_j \mathbb{I}_{\{j\in\mathcal{L}_t\}}$, where for an event X, $\mathbb{I}_X = 1$ if X holds, and $\mathbb{I}_X = 0$ otherwise. Then, the selected user is processed with the highest rate (Line 5). The process continues until no more user can be processed.

Theorem 2. WRAND *is* $(2 - \frac{1}{U})$*-competitive against an oblivious adversary.*

C. *Discussion*

We provide some remarks on the proposed algorithms.

- First, WFAIR and WRAND characterize the competitive ratio as a function of the scarcity level. The worst competitive ratio bound (equal to 2) for these algorithm, which occurs when U tends to infinity, matches the existing results with maximum charging rate [9], [17], [21]. In practice, however, the scarcity level is expected to be a small constant as the capacity is usually set based on the expected demand (as in, e.g., a cloud). Fig. 3 depicts the competitive ratio of the proposed algorithms against different values of U.
- The time complexity of WFAIR and WRAND are $O(n^2T)$ and $O(nT \log n)$, respectively. Thus, WRAND is a better choice in terms of computational complexity while attaining the same competitive ratio. Due to space constraints, we omit the details of the time complexity analysis.
- Finally, we mention that both our proposed algorithms are deadline-oblivious as they do not use the users' deadline in decision making. This property, on the one hand, proves useful in scenarios where the users' deadline are not provided to the system. It also makes the implementation easier. On the other hand, deadline-aware scheduling algorithms may enjoy a better a better competitive ratio (than that of deadline-oblivious ones) by utilizing the deadline information. Probably, no deadline-oblivious scheduling algorithm for RJSP can attain a competitive ratio better than $2 - \varepsilon$ for all $\varepsilon > 0$ when U grows large. An intuitive proof for this could be obtained by considering a scheduling problem in two time slots (i.e., $T = 2$) with $n + 2$ users, $C = K = 1$, and setting

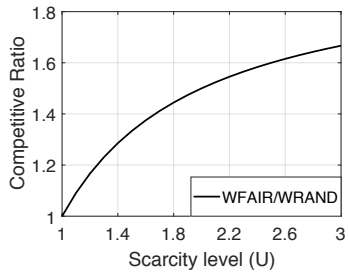


Fig. 3: Competitive ratio of proposed algorithms w.r.t. scarcity level (Definition 1).

$\pi_1 = \cdots = \pi_n = \langle\{1,2\},1,1\rangle$ and $\pi_{n+1} = \langle\{1\},1,1\rangle$. Since users 1 to $n+1$ only differ in their deadlines, they could not be distinguished by a deadline-oblivious algorithm. This would lead to a situation in which, with a high probability, user $n+1$ at $t=1$ would not be allocated any resource. In this case, the adversary can set $\pi_{n+2} = \langle\{2\},1,1\rangle$, thus resulting in a competitive ratio of 2.

V. Competitive Analysis

A. Preliminaries

The competitive analysis of our proposed algorithms relies on a proof technique, which is novel to the best of our knowledge. In this subsection, we describe our proof technique and illustrate it through some examples.

Let $\mathcal{A}$ be an online algorithm that outputs a feasible solution for RJSP. Let $y^{\mathcal{A}}_{j,t}$ denote the resource (charging rate) allocated to user j at time t under $\mathcal{A}$, and ALG be the corresponding objective value. Fix an optimal offline algorithm with objective value OPT and charging rates $y^{\star}_{j,t}, j \in \mathcal{N}, t \in \mathcal{T}$. If $y^{\mathcal{A}}_{j,t} \geq y^{\star}_{j,t}$ for all j and t, then $\mathcal{A}$ is optimal. However, if there exists a user j and a time slot t such that $y^{\star}_{j,t} > y^{\mathcal{A}}_{j,t}$, then the difference $y^{\star}_{j,t} - y^{\mathcal{A}}_{j,t}$ *might* increase the gap between ALG and OPT (by the amount $(y^{\star}_{j,t} - y^{\mathcal{A}}_{j,t})\rho_j$). Let $B_{j,t}$ be the *block*[3] of the resource that $\mathcal{A}$ allocated to user j at t with $|B_{j,t}| = y^{\mathcal{A}}_{j,t}$. Furthermore, denote by $\bar{B}_{j,t}$ the block corresponding to the additional resource that the optimal algorithm allocated to user j at t as compared to $\mathcal{A}$, which could be *feasibly* allocated by $\mathcal{A}$ to user j at t.

We denote by $\Phi_{j,t}$ the total gain that could be obtained by $\mathcal{A}$ if it had allocated $B_{j,t} \cup \bar{B}_{j,t}$ to j at time t. We have

$$\Phi_{j,t} = \rho_j y^{\mathcal{A}}_{j,t} + \bar{g}_{j,t}\,, \tag{3}$$

where $\bar{g}_{j,t}$ denotes the gain of $\mathcal{A}$ in block $\bar{B}_{j,t}$. To calculate $\bar{g}_{j,t}$, we need to know the valuation of EV(s) (if any) that occupied block $\bar{B}_{j,t}$ as well as the size of $\bar{B}_{j,t}$, which we denote by $\Delta_{j,t}$. Based on the previous discussion, $\Delta_{j,t}$ can be determined as follows:

$$\Delta_{j,t} = \begin{cases} \min\{y^{\star}_{j,t} - y^{\mathcal{A}}_{j,t}, R_{j,t}\} & y^{\star}_{j,t} \geq y^{\mathcal{A}}_{j,t} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

[3]Block is a conceptual term that facilitates our theoretical analysis and is not appeared in the main body of algorithm design.

The gain of $\mathcal{A}$ in sum of the two blocks $\bar{B}_{j,t}$ and $B_{j,t}$ is $\rho_j\Delta_{j,t}$ units less than that of the optimal solution unless $\mathcal{A}$ allocates the difference $\Delta_{j,t}$ to some other EVs and obtains the corresponding gain, $\bar{g}_{j,t}$. If $\mathcal{A}$ allocates the whole block $\bar{B}_{j,t}$ to a single EV i, then $\bar{g}_{j,t} = \Delta_{j,t}\rho_i$. More complex cases where there are more than one EV that occupy $\bar{B}_{j,t}$ will be considered later in the competitive analysis of our algorithms.

Next we introduce the notions of *gain* and *loss*. The gain of algorithm $\mathcal{A}$ at time t is defined as follows:

$$\Gamma_{\mathcal{A},t} = \sum_j \rho_j y^{\mathcal{A}}_{j,t}. \tag{5}$$

Furthermore, we note that $\text{ALG} = \sum_{t\in\mathcal{T}} \Gamma_{\mathcal{A},t}$.

Define $L_{\mathcal{A},t}$ as the *loss* of $\mathcal{A}$ at t expressed as

$$L_{\mathcal{A},t} = \text{OPT} - \text{OPT}^{-t}_{\mathcal{A}},$$

where $\text{OPT}^{-t}_{\mathcal{A}}$ is the optimal value of a variant of RJSP where the resource allocated to any user i at time t coincides to that allocated by $\mathcal{A}$. Equivalently, $\text{OPT}^{-t}_{\mathcal{A}}$ is the optimal value of RJSP with the following additional constraint: for all i, $y_{i,t} = y^{\mathcal{A}}_{i,t}$. Moreover, the total loss of $\mathcal{A}$ is given by $L_{\mathcal{A}} = \sum_{t\in\mathcal{T}} L_{\mathcal{A},t}$. The value $L_{\mathcal{A},t}$ characterizes the amount that $\mathcal{A}$ deviates from OPT at slot t. Define the loss of user j at slot t as

$$L_{j,t} = \rho_j\Delta_{j,t}, \quad t \in \mathcal{T}_j. \tag{6}$$

Then an upper bound on $L_{\mathcal{A},t}$ can be obtained as follows:

$$L_{\mathcal{A},t} \leq \sum_{j\in\mathcal{S}_{\text{OPT},t}} L_{j,t}.$$

In the following theorem, we relate the notion of loss of an algorithm $\mathcal{A}$ to its competitive guarantee.

Theorem 3. *If $L_{j,t} \leq c\Phi_{j,t}$ for all j and t, for some $c \geq 0$, then $\mathcal{A}$ is $(1+c)$-competitive.*

In what follows, we first define the notion of work-conserving algorithm and then provide two examples to illustrate the application of the technical tool described above.

Definition 2 (Work-Conserving Algorithm [33])**.** *A scheduling algorithm is work-conserving if it processes requests as long as there is some resources to allocate.*

Example 1: Consider a scheduling problem during 2 time slots ($T=2$) with $C=1$ and $K=1$ as shown in Fig. 1a. At the first time slot, users 1 and 2 arrive with demand profiles $\pi_1 = (\{1,2\},1,1)$ and $\pi_2 = (\{1\},1,1)$. Consider an algorithm $\mathcal{A}$ that selects user 1 to process at time slot 1. The gain at the first time slot is $\Gamma_{\mathcal{A},1} = 1$. For the second slot, we consider two scenarios as shown in Figs. 1b-1c. In the first scenario (Fig. 1b), where no EV arrives, we get $\text{OPT} = 2$ (by setting $\vec{y}^{\star}_1 = [0,1]$ and $\vec{y}^{\star}_2 = [1,0]$). Since $\mathcal{A}$ already fully charged EV 1, we get $\text{ALG} = 1$ (with $\vec{y}_1 = [1,0]$ and $\vec{y}_2 = [0,0]$). To obtain $\text{OPT}^{-1}_{\mathcal{A}}$ we fix algorithm $\mathcal{A}$'s decision at time slot 1 (that is selecting user 1) and find the maximum objective value that can be obtained by $\mathcal{A}$ which

is 1. Therefore, $\text{OPT}_{\mathcal{A}}^{-1} = 1$. The loss of $\mathcal{A}$ at $t = 1$ is then $L_{\mathcal{A},1} = \text{OPT} - \text{OPT}_{\mathcal{A}}^{-1} = 2 - 1 = 1$. Now let us consider the second scenario, where user 3 arrives at $t = 2$ with $\pi_3 = (\{2\}, 1, 1)$ (in Fig. 1c). In this case, if $\mathcal{A}$ sets $y_{3,2} = 1$, then $\Gamma_{\mathcal{A},2} = 1$ and thus $\Gamma_{\mathcal{A}} = 2$. We have $\text{OPT} = 2$ and so, $\text{OPT}_{\mathcal{A}}^{-1} = 2$ and $L_{\mathcal{A},1} = \text{OPT} - \text{OPT}_{\mathcal{A}}^{-1} = 2 - 2 = 0$.

Example 2 (Competitive analysis of FIRSTFIT *[11])*: FIRSTFIT [11] is a natural 2-competitive greedy scheduling algorithm that sorts the users based on their unit values and selects them one at a time until no more user can be allocated. Each time, the most valuable EV from the sorted list is selected and the processing rate is set to the maximum feasible rate. The algorithm continues until no more feasible allocation is possible (therefore, the algorithm is work-conserving). According to Theorem 3, the competitive ratio of FIRSTFIT is $(1 + \max_{j,t} L_{j,t}/\Phi_{j,t})$.

In what follows we apply our technique to derive the competitive ratio of FIRSTFIT. For selected user j at slot t by the optimal solution, we have $L_{j,t} \leq \rho_j y^\star_{j,t}$. If $\Delta_{j,t} = 0$, it means that the block $\bar{B}_{j,t}$ is allocated to some other users with at least the same unit values as j otherwise, FIRSTFIT would process j with a higher speed. Therefore, $\bar{g}_{j,t} \geq \rho_j \Delta_{j,t}$. Thus, noting that $y_{j,t} + \Delta_{j,t} = y^\star_{j,t}$, we get

$$\frac{L_{j,t}}{\Phi_{j,t}} = \frac{\rho_j \Delta_{j,t}}{\rho_j y_{j,t} + \bar{g}_{j,t}} \leq \frac{\rho_j \Delta_{j,t}}{\rho_j \Delta_{j,t}} = 1.$$

Applying Theorem 3 proves that FIRSTFIT is 2-competitive.

We conclude this subsection by the following definition:

Definition 3 (Saturated Time Slot)**.** *A time slot t is said to be* saturated *if it satisfies $\sum_j y_{j,t} = C$.*

B. WFAIR *Analysis (Proof of Theorem 1)*

We first note that we assumed $U > 1$. For the case where $U \leq 1$, it is trivial to show that WFAIR is optimal as there will always be sufficient resources to schedule all users with the maximum speed.

To prove the theorem, we compute $L_{j,t}$ and $\Phi_{j,t}$ for WFAIR, and then apply Theorem 3. Without loss of generality, assume $0 < \rho_i \leq 1$ for all i and $\sum_{i \in \mathcal{M}_t} \rho_i = 1$. This is always possible through normalization, namely by dividing the unit value of each user to the sum of unit values of all active users. Moreover, we assume that for any active job $i \neq j$ at time t, it holds that $R_{i,t} \leq K$. This assumption can be relaxed by temporarily aggregating multiple demands into a single demand for the current slot and then splitting them at the next slot.

Let $\mathcal{A} =$ WFAIR, and to ease notation, in the rest of the proof we omit the dependence of $y^{\mathcal{A}}_{j,t}$ on $\mathcal{A}$ for all j and t (so $y_{j,t} := y^{\mathcal{A}}_{j,t}$). Fix an optimal solution and a user $j \in \mathcal{S}_{\text{OPT},t}$. Let $y^\star_{j,t}$ denote the amount of resource allocated by the optimal solution to j at time t. Since we consider the worst-case, in the rest of the proof we assume that j is not completed by WFAIR (otherwise, $L_{j,t} = 0$) and thus, $j \in \mathcal{M}_t$.

If $\sum_{i \in \mathcal{M}_t} \min\{K, R_{i,t}\} \leq C$, then all active users can be scheduled with the maximum feasible rate at t and the gain is $\rho_i \min\{K, R_{i,t}\}$ for all $i \in \mathcal{M}_t$. In this case, $L_{\mathcal{A},t} = 0$ since for any available user i, $y_{i,t} = \min\{K, R_{i,t}\} \geq y^\star_{j,t}$ or i is completed in an earlier time slot.

Now, we focus on the case where $\sum_{j \in \mathcal{M}_t} \min\{K, R_{i,t}\} > C$. First, observe that WFAIR is work-conserving since the "while" loop in WFAIR will not terminate if more resources can be allocated to the users. We further deduce that time slot t is saturated. This implies that there must be a non-empty set $\mathcal{H} \subseteq \mathcal{N}_t \backslash \{j\}$ of users such that they received the difference $\Delta_{j,t}$. Let $H = |\mathcal{H}|$ and note that $H \leq n_t - 1$. Let $y_{i,t}, i = 1, \ldots, H$ be the amount that WFAIR allocated to user $i \in \mathcal{H}$ with

$$\Delta_{j,t} = \sum_{i \in \mathcal{H}} y_{i,t}$$

Then, we have $\bar{g}_{j,t} = \sum_{i \in \mathcal{H}} \rho_i y_{i,t}$. According to allocation strategy of WFAIR, $y_{j,t} = \min\{\rho_j C, K, R_{j,t}\}$. Since $\Delta_{j,t} > 0$, thus, $y_{j,t} < K$. Also, as j is not yet finished by WFAIR at t, $y_{j,t} < R_{j,t}$. Therefore, $y_{j,t} = \rho_j C$. Since $y_{i,t} \leq \rho_i C$ for all $i \in \mathcal{H}$. Thus,

$$\sum_{i \in \mathcal{H}} \rho_i y_{i,t} \geq \frac{1}{C} \sum_{i \in \mathcal{H}} y^2_{i,t},$$

which further gives $\bar{g}_{j,t} \geq \frac{1}{C} \sum_{i \in \mathcal{H}} y^2_{i,t}$. The right-hand side of the above is minimized with $y_{i,t} = \frac{1}{H} \Delta_{j,t}$. Hence,

$$\bar{g}_{j,t} \geq \frac{1}{C} \sum_{i \in \mathcal{H}} \frac{\Delta^2_{j,t}}{H^2} = \frac{\Delta^2_{j,t}}{CH},$$

and we get

$$\frac{L_{j,t}}{\Phi_{j,t}} \leq \frac{\rho_j \Delta_{j,t}}{\rho_j^2 C + \frac{\Delta^2_{j,t}}{CH}}.$$

Let $\Delta_{j,t} = a\rho_j$ where $a > 0$ is a constant to be identified. By replacing $\Delta_{j,t}$ we have $\frac{L_{j,t}}{\Phi_{j,t}} \leq \frac{aCH}{C^2 H + a^2}$. The maximum value of this term is obtained by setting $a = C$. Therefore,

$$\frac{L_{j,t}}{\Phi_{j,t}} \leq \frac{C^2 H}{C^2 H + C^2} = 1 - \frac{1}{H+1}.$$

It just remains to find an upper bound for H. To this end, we define the notion of *importance ratio*.

Definition 4 (Importance Ratio [13])**.** *Given a set $\mathcal{M}$ of users, the* importance ratio *of $\mathcal{M}$ is defined as the maximum ratio of unit values of users in $\mathcal{M}$, i.e., $r_{\mathcal{M}} := \max_{i,j \in \mathcal{M}} \frac{\rho_j}{\rho_i}$.*

In this paper, we assume that the importance ratio does not grow with the number n of users. We have:

Lemma 1. *Assume that at each time slot t, the unit values of active users are normalized and add up to 1. Then,*

$$\rho_j \geq \frac{1}{n + r_{\mathcal{M}_t} - 1}, \quad \forall j \in \mathcal{M}_t.$$

Having $y_{j,t} + \sum_{i=1}^{H} y_{i,t} \leq K$ and $R_{i,t} \geq K$ for all $i \in \mathcal{H}$, we get $y_{i,t} = \rho_i C$ for all $i \in \mathcal{H}$. Using Lemma 1, we get $\frac{HC}{n + r_{\mathcal{M}_t} - 1} + \rho_j C \leq K$, thus giving

$$H \leq \frac{K(n + r_{\mathcal{M}_t} - 1) - C}{C} \approx \frac{nK - C}{C} = U - 1.$$

Here, we made the approximation based on the fact that $r_{\mathcal{M}_t}$ is a constant and n is large. Therefore,

$$\frac{L_{j,t}}{\Phi_{j,t}} \leq \frac{U-1}{U}.$$

Finally, applying Theorem 3 we conclude that WFAIR is $(2-\frac{1}{U})$-competitive. □

C. WRAND *Analysis (Proof of Theorem 2)*

To analyze competitive ratio of WRAND, we assume an *oblivious adversary* model [34], which is reasonable in practical scenarios. Recall that an oblivious adversary has complete knowledge about the algorithm's code but has no information about the random choices made by the algorithm during its execution.

Let $\mathcal{A} = \text{WRAND}$ and for brevity, in the rest of the proof, omit the dependence of $y^{\mathcal{A}}_{j,t}$ on $\mathcal{A}$ for all j and t (so $y_{j,t} := y^{\mathcal{A}}_{j,t}$). Fix an optimal solution, and consider a user j and a time slot t such that $j \in \mathcal{S}_{\text{OPT},t}$. Without loss of generality, we make the following assumptions:

(i) Using a similar argument as in the proof of Theorem 1 and to consider a worst-case scenario, we assume that j is not completed by WRAND at t and slot t is saturated.
(ii) We assume that $y^\star_{j,t} > y_{j,t}$, since otherwise $L_{j,t} = 0$.
(iii) If $j \in \mathcal{S}_{\mathcal{A},t}$, then $R_{j,t} > y^\star_{j,t}$. This is because in the otherwise case (i.e., $R_{j,t} \leq y^\star_{j,t}$), we get $R_{j,t} \leq K$ and considering the fact that WRAND allocates the maximum feasible resource to selected users, then j should be finished at t and thus $\Delta_{j,t} = 0$ and subsequently $L_{j,t} = 0$.

Given that $R_{j,t} > y^\star_{j,t}$, we get $\Delta_{j,t} = \min\{R_{j,t}, y^\star_{j,t} - y_{j,t}\} = y^\star_{j,t} - y_{j,t}$, and using Eq. (6), we have

$$L_{j,t} = \begin{cases} \rho_j y^\star_{j,t} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ 0 & j \in \mathcal{S}_{\mathcal{A},t}, \end{cases} \tag{7}$$

thus giving

$$\mathbb{E}[L_{j,t}] = \Pr(j \notin \mathcal{S}_{\mathcal{A},t}) \rho_j y^\star_{j,t}.$$

If $j \notin \mathcal{S}_{\mathcal{A},t}$, the algorithm prefers another user, name i, with occupied block $\bar{B}_{j,t}$. Note that the case that $\bar{B}_{j,t}$ is allocated to more than one user does not affect the analysis of WRAND as in this case the weighted average of users' unit value can be considered. Therefore, the gain $\bar{g}_{j,t} = \rho_i y^\star_{j,t}$ and $\rho_j y_{j,t} = 0$. On the other hand, when $j \in \mathcal{S}_{\mathcal{A},t}$, since $\Delta_{j,t} = 0$, then $\bar{g}_{j,t} = 0$. Therefore, using Eq. (3), we can calculate $\Phi_{j,t}$ as

$$\Phi_{j,t} = \begin{cases} \rho_i y^\star_{j,t} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ \rho_j y^\star_{j,t} & j \in \mathcal{S}_{\mathcal{A},t}, R_{j,t} \geq y^\star_{j,t}, \\ \rho_j R_{j,t} & j \in \mathcal{S}_{\mathcal{A},t}, R_{j,t} < y^\star_{j,t}. \end{cases}$$

Using (iii), we ignore the third case and so,

$$\Phi_{j,t} \geq \begin{cases} \rho_i y^\star_{j,t} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ \rho_j y^\star_{j,t} & j \in \mathcal{S}_{\mathcal{A},t}. \end{cases} \tag{8}$$

Let $h : \mathcal{S}_{\text{OPT},t} \times \mathcal{T} \to \mathbb{R}_+$ be a function with $h(j',t') = L_{j',t'}/\Phi_{j',t'}$ if $j' \notin \mathcal{S}_{\text{OPT},t}$ and $h(j',t') = 0$, otherwise. Note that $\Phi_{j',t'} > 0$ as unit values are positive numbers. Using Eqs. (7) and (8), we obtain

$$h(j,t) \leq \begin{cases} \frac{\rho_j}{\rho_i} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ 0 & j \in \mathcal{S}_{\mathcal{A},t}, \end{cases}$$

and thus,

$$\mathbb{E}[h(j,t)] \leq \frac{\rho_j}{\rho_i} \Pr(j \notin \mathcal{S}_{\mathcal{A},t}).$$

By the design of WRAND, the selection probabilities are proportional to the unit values. Hence,

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) = \frac{\rho_i}{\rho_j} \Pr(i \notin \mathcal{S}_{\mathcal{A},t}).$$

Lemma 2. *Let t be a saturated time slot. Then, under $\mathcal{A} =$* WRAND*,*

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) \leq 1 - \frac{1}{U}, \quad \forall j \in \mathcal{M}_t.$$

Now applying Lemma 2 gives

$$\mathbb{E}[h(j,t)] \leq \frac{\rho_j}{\rho_i} \frac{1}{\rho_j/\rho_i} \Pr(i \notin \mathcal{S}_{\mathcal{A},t}) \leq 1 - \frac{1}{U}.$$

Applying Theorem 3, we finally conclude that WRAND is $(2-\frac{1}{U})$-competitive. □

VI. Simulation Results

In this section, we evaluate the *average performance* of the proposed algorithms. Although we provided theoretical bounds for the worst-case performance of our methods, the average case performance is still important. We note that it is possible that an algorithm with poor competitive ratio can beat another algorithm with a good competitive ratio in the average case.

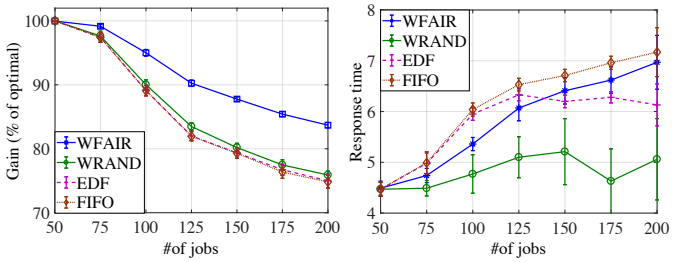
A. Setup

The default parameter setting, unless otherwise mentioned, is as follows: we consider a charging station in a time period of 16 hours with $T = 16$. The resource constraint at each time slot is 200 kWh. Similarly to [17], we assume that the number of arrivals at each time slot follows a Poisson distribution with a mean of 10. Moreover, the length of availability window of an EV is independent from the others and follows an exponential distribution. For each EV, the maximum charging rate is drawn uniformly at random from the interval $[1, 10]$. The demand of each EV j is sampled uniformly at random from the interval $[\frac{1}{3}K|\mathcal{T}_j|, K|\mathcal{T}_j|]$, and the value v_j is sampled uniformly from the interval $[\frac{1}{2}D_j, 5D_j]$.

We used Gurobi solver [35] to find the optimal solution and compare the performance of WFAIR and WRAND to the optimal offline solution as well as two other benchmarks:

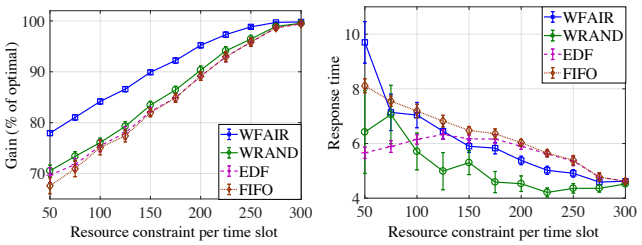
- *FIFO (First In First Out)*: At each time slot, the priority is given to the EVs with earlier arrival time.
- *EDF (Earliest Deadline First)*: At each time slot, the priority is given to the EVs that are closer to their deadline.

Two major metrics are studied in the simulation: a) the gain of the system which is identified by the objective function in RJSP, and b) average response time of the EVs, defined as



(a) Gain (b) Response time

Fig. 4: Varying number of EVs.



(a) Gain (b) Response time

Fig. 5: Varying resource constraint.

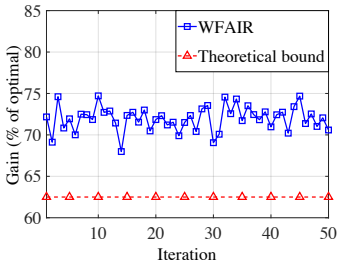


Fig. 6: Tracking worst-case performance in 50 random scenarios for WFAIR.

the average number of time slots to complete EVs' demand. To compute the response time, we only considered EVs who received their total demand and ignored partially charged EVs.

B. Impact of Number of Users

In the first scenario in Fig. 4, the number of EVs is changed from 50 to 200 while the other parameters are set to their default values as described in Section VI-A. When the number of EVs is small, the scheduling problem is less challenging. As it can be seen in Fig. 4a, for $n = 50$ the gain of all methods is close to the optimal one. As n increases, the gain falls down for all algorithms. On average, WFAIR has the best performance by achieving 91% of the optimal while the difference between WRAND, EDF and FIFO is minuscule (86%, 85% and 85% of the optimal, respectively). The average response time of all methods increases by increasing number of EVs, where WFAIR and FIFO show more sensitivity to this change. Another observation is that the response time of WFAIR is higher than WRAND while this is reverse for the gain in Fig. 4a.

C. Impact of Resource Constraint

In the second scenario, the resource constraint at each time slot, C, is varied from 50 kWh to 300 kWh and the methods are compared based on their gain and average response time. By increasing the resource constraint, it is expected that both gain and response time of the algorithms improve. The reason is that with more available resources, EVs have not to wait too long to be allocated. Besides, more EVs can be served at each time slot. This is observable in Figs. 5a and 5b. Similarly to the scenario of Fig. 4, WFAIR outperforms WRAND in terms of gain while there is a small gap between WRAND, EDF, and FIFO. When the available resource is sufficient to easily serve all EVs (at $C = 300$), the gain of all methods converges to the optimal gain.

D. Confirming the Theoretical Bounds

The analysis in Section IV-A demonstrates that the performance of WFAIR should not fall down its competitive ratio under any input scenario. To verify, we generated 50 random scenarios with $n = 100$, $C = 200$ and set $K = 5$ which gives $U = 2.5$ and the competitive ratio of 1.6. Then, we compared the WRAND algorithm with the optimal solution in each single scenario and plotted the result in Fig. 6. It can be observed that the gain of WFAIR is *always* significantly better than the worst-case gain suggested by the competitive ratio. The simulation result here confirms the theory.

VII. Conclusion

This study set out to tackle deadline constrained job scheduling problem with its application in electric vehicles. Two deadline-oblivious online algorithms (one deterministic and one randomized) have been developed and their performance are analyzed by a new proof technique, which can be used to find upper bound for competitive ratio of a class of algorithms designed for the studied problem. Under realistic scenarios where the demand-to-supply ratio is not too high, the proposed algorithms improve the state of the art result. Further research should be conducted on the scheduling algorithms which can utilize deadline information of the users. Moreover, the proposed proof technique could be extended to support a wider range of the problems.

References

[1] F. Li, "Competitive scheduling of packets with hard deadlines in a finite capacity queue," in *IEEE INFOCOM*, pp. 1062–1070, 2009.

[2] L. Yang, W. S. Wong, and M. H. Hajiesmaili, "An optimal randomized online algorithm for QoS buffer management," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 36:1–36:26, 2017.

[3] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," in *USENIX OSDI*, 1994.

[4] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *IEEE FOCS*, pp. 374–382, 1995.

[5] Y. Zheng, *Scheduling and design in cloud computing systems*. PhD Thesis, The Ohio State University, 2015.

[6] Y. Zhou, D. Yau, P. You, and P. Cheng, "Optimal-cost scheduling of electrical vehicle charging under uncertainty," *to appear in IEEE Transactions on Smart Grid*.

[7] S. Chen, Y. Ji, and L. Tong, "Large scale charging of electric vehicles," in *IEEE PESGM*, pp. 1–9, 2012.
[8] J. C. Mukherjee and A. Gupta, "A review of charge scheduling of electric vehicles in smart grid," *IEEE Systems Journal*, vol. 9, no. 4, pp. 1541–1553, 2015.
[9] N. Jain, I. Menache, J. S. Naor, and J. Yaniv, "Near-optimal scheduling mechanisms for deadline-sensitive jobs in large computing clusters," *ACM Transactions on Parallel Computing*, vol. 2, no. 1, pp. 3:1–3:29, 2015.
[10] Y. Azar, I. Kalp-Shaltiel, B. Lucier, I. Menache, J. S. Naor, and J. Yaniv, "Truthful online scheduling with commitments," in *ACM EC*, pp. 715–732, 2015.
[11] E.-C. Chang and C. Yap, "Competitive on-line scheduling with level of service," *Journal of Scheduling*, vol. 6, no. 3, pp. 251–267, 2003.
[12] F. Y. Chin and S. P. Fung, "Improved competitive algorithms for online scheduling with partial job values," in *COCOON*, pp. 425–434, 2003.
[13] F. Y. Chin and S. P. Fung, "Online scheduling with partial job values: Does timesharing or randomization help?," *Algorithmica*, vol. 37, no. 3, pp. 149–164, 2003.
[14] M. Chrobak, L. Epstein, J. Noga, J. Sgall, R. van Stee, T. Tichỳ, and N. Vakhania, "Preemptive scheduling in overloaded systems," *Automata, Languages and Programming*, pp. 800–811, 2002.
[15] Y. He, Z. Ye, Q. Fu, and S. Elnikety, "Budget-based control for interactive services with adaptive execution," in *ACM ICAC*, pp. 105–114, 2012.
[16] Y. Zheng, B. Ji, N. Shroff, and P. Sinha, "Forget the deadline: Scheduling interactive applications in data centers," in *IEEE CLOUD*, pp. 293–300, 2015.
[17] Z. Zheng and N. B. Shroff, "Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud," in *IEEE INFOCOM*, pp. 1–9, 2016.
[18] Y. He, S. Elnikety, J. Larus, and C. Yan, "Zeta: Scheduling interactive services with partial execution," in *ACM SoCC*, 2012.
[19] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. Cambridge University Press, 1998.
[20] S. Chen, L. Tong, and T. He, "Optimal deadline scheduling with commitment," in *Allerton*, pp. 111–118, 2011.
[21] B. Lucier, I. Menache, J. S. Naor, and J. Yaniv, "Efficient online scheduling for deadline-sensitive jobs," in *ACM SPAA*, pp. 305–314, 2013.
[22] M. T. Hajiaghayi, "Online auctions with re-usable goods," in *ACM EC*, pp. 165–174, 2005.
[23] A. Gupta, R. Krishnaswamy, and K. Pruhs, "Online primal-dual for non-linear optimization with applications to speed scaling," in *WAOA*, pp. 173–186, 2012.
[24] W. Tang, S. Bi, and Y. J. Zhang, "Online charging scheduling algorithms of electric vehicles in smart grid: An overview," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 76–83, 2016.
[25] R. Deng and H. Liang, "Whether to charge an electric vehicle or not? A near-optimal online approach," in *IEEE PESGM*, pp. 1–5, 2016.
[26] W. Tang, S. Bi, and Y. J. Zhang, "Online speeding optimal charging algorithm for electric vehicles without future information," in *IEEE SmartGridComm*, pp. 175–180, 2013.
[27] W. Tang, S. Bi, and Y. J. A. Zhang, "Online coordinated charging decision algorithm for electric vehicles without future information," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2810–2824, 2014.
[28] E. H. Gerding, V. Robu, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings, "Online mechanism design for electric vehicle charging," in *AAMAS*, pp. 811–818, 2011.
[29] 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 *AMMA*, pp. 100–112, 2011.
[30] 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*, vol. 48, pp. 175–230, 2013.
[31] S. Chen and L. Tong, "iems for large scale charging of electric vehicles: Architecture and optimal online scheduling," in *IEEE SmartGridComm*, pp. 629–634, 2012.
[32] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 23:1–23:26, 2017.
[33] K. Funaoka, S. Kato, and N. Yamasaki, "Work-conserving optimal real-time scheduling on multiprocessors," in *ECRTS*, pp. 13–22, 2008.
[34] R. Motwani and P. Raghavan, *Randomized algorithms*. Chapman & Hall/CRC, 2010.
[35] "Gurobi optimizer 5.0," *Gurobi: http://www.gurobi.com*, 2013.

APPENDIX

A. Proof of Lemma 1

Let $\rho_{\max,t}$ and $\rho_{\min,t}$ denote the maximum and minimum unit values at t, respectively. Recall that by definition, $r_{\mathcal{M}_t} = \frac{\rho_{\max,t}}{\rho_{\min,t}}$. To prove the lemma, it suffices to derive a lower bound on $\rho_{\min,t}$. Observe that the minimal value of $\rho_{\min,t}$ occurs when there are $n_t - 1$ users with unit value $\rho_{\min,t}$ and one user with $\rho_{\max,t}$. It then follows that

$$(n_t - 1)\rho_{\min,t} + r_{\mathcal{M}_t}\rho_{\min,t} = 1,$$

since unit values are normalized. Using $n_t \leq n$ gives the desired result. □

B. Proof of Lemma 2

Let $j \in \mathcal{M}_t$ and consider a saturated time slot t. First note that by definition, at least C/k users are selected in time slot t (i.e., $n_t \geq C/k$), where at each round the selected users in previous rounds are excluded from the selection pool. This can be modeled as a hypergeometric distribution with n_t balls, where $\rho_j n_t$ of them are of our interest. Furthermore, the number of draws is C/k and we will succeed if at least one of those $\rho_j n_t$ balls are selected (to simplify the presentation, we assume that $\rho_j n_t, C/k \in \mathbb{N}$). It then follows that the probability that user j is not selected is given by:

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) = \frac{\binom{n_t - \rho_j n_t}{C/k}}{\binom{n_t}{C/k}}.$$

Using Lemma 1, $\rho_j \geq \frac{1}{n + r_{\mathcal{M}_t} - 1} \approx \frac{1}{n}$. Moreover, $\frac{n_t}{U_t} = \frac{C}{k}$ so that

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) \approx \frac{\binom{n_t - 1}{n_t/U_t}}{\binom{n_t}{U_t}} = 1 - \frac{1}{U_t} \leq 1 - \frac{1}{U},$$

which concludes the proof. □

C. Proof of Theorem 3

To prove the theorem, we provide lower and upper bounds on $L_{\mathcal{A}}$. First observe from the definition of $L_{\mathcal{A},t}$ that the gap between OPT and $\Gamma_{\mathcal{A}}$ is less than or equal to the aggregate loss over the time horizon:

$$\text{OPT} - \text{ALG} \leq L_{\mathcal{A}}.$$

On the other hand, we have

$$\begin{aligned} L_{\mathcal{A}} = \sum_{t \in \mathcal{T}} L_{\mathcal{A},t} &\leq \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{S}_{\text{OPT},t}} L_{j,t} \\ &\leq \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{S}_{\text{OPT},t}} c\Phi_{j,t} \\ &= c \sum_{t \in \mathcal{T}} \Gamma_{\mathcal{A},t} = c\text{ALG}. \end{aligned}$$

Putting the two bounds together gives $\text{OPT} \leq (1+c)\text{ALG}$ and completes the proof. □