Fair resource allocation in virtualized O-RAN platforms

Aslan, Fatih; Iosifidis, George; Ayala-Romero, Jose A.; Garcia-Saavedra, Andres; Costa-Perez, Xavier

DOI
10.1145/3639043

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the ACM on Measurement and Analysis of Computing Systems

Fair Resource Allocation in Virtualized O-RAN Platforms

FATIH ASLAN and GEORGE IOSIFIDIS, TU Delft, The Netherlands
JOSE A. AYALA-ROMERO and ANDRES GARCIA-SAAVEDRA, NEC Laboratories Europe, Germany
XAVIER COSTA-PEREZ, i2CAT, NEC Laboratories Europe and ICREA, Spain

O-RAN systems and their deployment in virtualized general-purpose computing platforms (O-Cloud) constitute a paradigm shift expected to bring unprecedented performance gains. However, these architectures raise new implementation challenges and threaten to worsen the already-high energy consumption of mobile networks. This paper presents first a series of experiments which assess the O-Cloud's energy costs and their dependency on the servers' hardware, capacity and data traffic properties which, typically, change over time. Next, it proposes a compute policy for assigning the base station data loads to O-Cloud servers in an energy-efficient fashion; and a radio policy that determines at near-real-time the minimum transmission block size for each user so as to avoid unnecessary energy costs. The policies balance energy savings with performance, and ensure that both of them are dispersed fairly across the servers and users, respectively. To cater for the unknown and time-varying parameters affecting the policies, we develop a novel online learning framework with fairness guarantees that apply to the entire operation horizon of the system (long-term fairness). The policies are evaluated using trace-driven simulations and are fully implemented in an O-RAN compatible system where we measure the energy costs and throughput in realistic scenarios.

CCS Concepts: • **Networks** → **Network performance evaluation**; • **Theory of computation** → **Online learning algorithms**.

Additional Key Words and Phrases: Online Learning, Regret, Mobile Networks, O-RAN, Fairness, Resource Management, Energy Efficiency

ACM Reference Format:
Fatih Aslan, George Iosifidis, Jose A. Ayala-Romero, Andres Garcia-Saavedra, and Xavier Costa-Perez. 2024. Fair Resource Allocation in Virtualized O-RAN Platforms. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 1, Article 17 (March 2024), 34 pages. https://doi.org/10.1145/3639043

1 INTRODUCTION

1.1 Background & Motivation

One of the most revolutionizing aspects of future mobile networks is the *virtualization* of the Radio Access Network (vRAN), in particular of the base stations (vBS), and the execution of their software functions at general-purpose computing platforms [28]. Driven by the Open RAN (O-RAN) Alliance, practically the entire Telco industry is currently investing in the development of vBSs, in anticipation of the eclipse of conventional RANs by 2028 [8]. Virtualized RANs promote the control of vBSs in (almost) real-time, using new knobs that tailor their operation to the environment, e.g., channel conditions, and to user needs for throughput, latency, and other KPIs. The proposed vRAN architectures typically include computing pools (O-Cloud) of heterogeneous processing units (PUs),

Authors' addresses: Fatih Aslan, f.aslan@tudelft.nl; George Iosifidis, g.iosifidis@tudelft.nl, TU Delft, The Netherlands; Jose A. Ayala-Romero, jose.ayala@neclab.eu; Andres Garcia-Saavedra, andres.garcia.saavedra@neclab.eu, NEC Laboratories Europe, Germany; Xavier Costa-Perez, xavier.costa@i2cat.net, i2CAT, NEC Laboratories Europe and ICREA, Spain.

ACM 2476-1249/2024/3-ART17
https://doi.org/10.1145/3639043

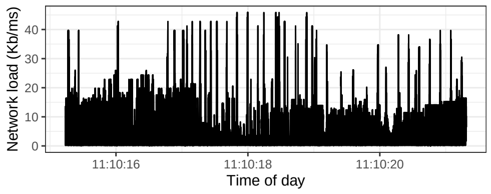


Fig. 1. Cell load dynamics (msec granularity) over a few seconds, collected from an operational RAN in Frankfurt, Germany, May 2023.

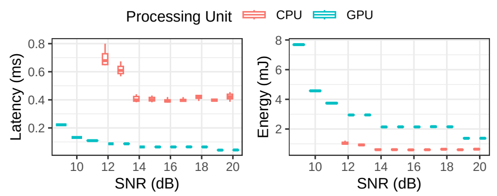


Fig. 2. Processing latency (left) and energy consumption (right) to process one TB under different SNRs; measured on an Intel Xeon CPU core and an NVIDIA V100 GPU.

with CPUs or ASIC/FPGA/GPU hardware accelerators (HAs), which execute dynamically-allocated compute workloads of one or more vBSs [73]. This native cloud-based architecture constitutes a paradigm shift for RAN and is anticipated to bring unprecedented performance gains [42].

Unfortunately, the virtualization of RAN is expected also to increase the Operating Expenditures (OpEx) of networks due to the high energy consumption of vBSs. Namely, unlike legacy base stations, the energy spent for executing the software vBS functions becomes very relevant and, in fact, can even surpass that of wireless transmissions [11, 18]. Moreover, these costs are volatile and unpredictable, as they depend on a range of factors such as the radio characteristics of the transmitted data (e.g., the Signal-to-Noise Ratio, SNR), and the properties of the O-Cloud PUs. Coupled with the increasing RAN densification, this effect is bound to render the vRAN energy costs — an already prevalent concern for operators[1] — prohibitively high for future mobile networks. Indeed, there is wide consensus that this is a key obstacle hampering the adoption of vRANs [7], and hence is justifiably very high in the O-RAN agenda of industries [34, 84].

A promising method to tackle this issue is to leverage one of the key O-RAN architecture innovations: the RAN Intelligent Controller. The RIC, as commonly termed, provides a centralized abstraction of the network and is envisioned as a powerful enabler for control policies with different objectives, decision granularity, and time-scales [33, 42]. Interestingly, the RIC policies can shape the performance and energy cost of the vRAN in two ways: *(i)* by assigning carefully the vBSs workloads to different PUs of O-Cloud; and *(ii)* by affecting the characteristics of these workloads in almost real-time. For instance, the RIC could dictate the vBSs to route their most voluminous flows to servers equipped with HAs; to refrain from using energy-costly modulation schemes [16, 51]; or to bound their transmission power [15]. Such *compute control* and *radio control* policies can, in principle, be very effective in balancing the vRAN performance and energy costs, but require access to system and user parameters that are unknown and vary rapidly, and presume solving large-scale challenging optimization problems.

To exemplify, the processing time for decoding/encoding the users' uplink/downlink streams depends on the amount of this traffic which, most often, is subject to rapid fluctuations; see, e.g., [38] and our measurements in Fig. 1. Secondly, the processing time for each Transport Block (TB)[2] depends on its SNR, which might change drastically as users move around. To illustrate this, we show in Fig. 2 experimental results obtained using the testbed platform described in Sec. 6. Fig. 2 (left) presents the processing time of a TB (of certain length) at a CPU or GPU, under different SNRs[3]. Leveraging its high degree of data parallelization (intrinsic to its architecture) the GPU speeds up the TB processing, in contrast to the CPU where the processing is sequential (see [35] for more

[1]For instance, Verizon and Vodafone announced their target for net zero energy emissions by 2040 [10], and China Mobile has set to reduce energy consumption and carbon emissions by 20% in the next few years [56].
[2]TB is the basic MAC-layer user data unit, and its length (bits) depends typically on the radio resource blocks and MCS.
[3]5G FEC is implemented via LDPC iterative algorithms, which require more computations for lower SNR, see [26].

details). This delay variance is critical, as the vBS workloads are subject to stringent processing deadlines (1-3 ms) which, if violated, lead to data loss and energy waste [38, 40]. Third, the energy cost of these computations depends on the PU technology and the data stream characteristics. For example, the experiments in Fig. 2 (right) show that a GPU decodes a TB (with SNR 14 dB) approximately 5× faster than a CPU, but consumes 2.5× more energy; and demonstrate how the (unknown and varying) SNR affects this comparison. Motivated by these observations, this work introduces and evaluates an algorithmic toolbox for the design of *data-adaptive RIC policies*, in different time-scales, towards taming the energy consumption of vRANs while accounting for fairness criteria w.r.t. performance (across users) and energy costs (across servers).

1.2 Methods & Contributions

We focus on two key resource management problems that affect the vRANs' performance and energy, and design a computing control and a radio control policy to tackle them. The first problem studies the assignment of vBS workloads to O-Cloud processing units. The workloads differ in their volume and SNR, and similarly the PUs are heterogeneous in terms of technology (CPU or HA), capacity and energy consumption. We find experimentally that CPU-based PUs can process small and *cleaner* (high SNR) workloads with less energy; while GPU-based PUs can be used for voluminous and/or low-SNR data to ensure their timely processing. Therefore, we argue that a RIC at non-Real-Time (non-RT) can devise an intelligent workload assignment policy which dictates how the vBSs can leverage the PUs' diversity to balance energy costs and performance (successfully processed loads). Such a policy needs to adapt to the time-varying properties of workloads and PUs; and allocate fairly in the long-run the O-Cloud capacity across the vBSs, and the energy costs across the PUs. This latter property increases reliability (via load balancing), and is key for multi-vendor O-Clouds where the PUs are owned by different business entities.

The second problem concerns a new radio control policy, similar in flavor to those in [14–16]. The starting point here is our experiments (Sec. 6) showing that HAs consume almost the same energy per TB independently of its length[4]. Thus, if users transmit larger TBs, the vRAN will consume less energy per bit. This creates an opportunity for the RIC to introduce a highly dynamic (i.e., at near-RT scale) minimum TB size policy (`minTB`) for each user, preventing transmission of small TBs. Nevertheless, such a policy will inevitably deteriorate the latency for users, as they might need to refrain from transmitting despite having non-empty (MAC-layer) buffers. It is therefore imperative to strike a balance between the energy savings and transmission delays; and further, to disperse fairly these delays across the users so as to avoid excessive service deterioration for some of them. Deciding the `minTB` thresholds requires access to the user traffic and HA energy costs, which change with time and are typically unknown when such near-RT policies are devised. Further, it involves solving large-scale optimization problems in almost real-time scales.

We design a novel optimization toolbox for the above compute and radio control policies based on online learning, cf. [46, 79]. Our approach relies on the celebrated Follow The Regularized Leader (FTRL) framework [80], that has been particularly successful in the design of data-adaptive and robust decision policies [58]. We extend FTRL here to account for the specifics of these problems, namely we equip it with a *two-sided long-term* fairness metric, so as to support fairness w.r.t. cost savings across the servers and fairness w.r.t. performance gains across users over its entire operation; and we include predictions for the unknown (system and user) parameters. Achieving fairness in such dynamic decision models is technically challenging, and previous works are confined to per-slot fairness (which impacts efficiency), with only few exceptions, e.g., [44, 55, 81]. We overcome this barrier through a saddle-point transformation where in the dual space we track the two fairness

[4]GPUs have Streaming Multiprocessors that parallelize workloads effectively, and LDCP codes are amenable to parallelization.

metrics. The predictions, on the other hand, bring in the optimistic learning aspect [66, 76], and, if used judiciously, can expedite the learning rate when they are (relatively) accurate, without deteriorating it otherwise. The proposed algorithms offer optimality guarantees (i.e., *regret*) w.r.t. ideal benchmark policies one could only devise using oracles, and which hold for a wide range of perturbation models including adversarial ones. Our contributions are thus summarized as follows:

• We present new experimental results for the computing delay and energy costs in O-RAN, which motivate the design of dynamic and adaptive compute and radio control policies.

• We develop a lightweight learning framework for designing control policies which: *(i)* assign fairly the vBSs' compute workloads to O-Cloud processing units; and *(ii)* decide TB size thresholds, towards reducing the vRAN energy costs while ensuring fair performance across users.

• We prove the policies have sublinear regret under adversarial scenarios and assess their implementation overheads and dependency on system parameters, user demands and predictions accuracy. Along the road we develop technical results that improve these learning techniques.

• We evaluate the policies using trace-driven and synthetic simulations, and we implement them at an O-RAN-compliant testbed to measure the actual vBS performance and PUs energy costs.

Notation. $\|\cdot\|$, $\|\cdot\|_\infty$, and $\|\cdot\|_1$ denote the ℓ_2 (Euclidean), ℓ_∞ and ℓ_1 norms. Vector transpose and the Hadamard product are denoted $\top$ and $\circ$, respectively. We denote vectors with small bold typeface letters and use subscripts to index them. We use $\boldsymbol{x}_{1:t}$ for the sum of vectors $\sum_{i=1}^{t} \boldsymbol{x}_i$, and do so also for scalars. $\boldsymbol{x}_t$ time-indexes vector $\boldsymbol{x}$, and $\{\boldsymbol{x}_t\}_{t=1}^{T}$ denotes the sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T$. When the horizon is not relevant, we write $\{\boldsymbol{x}_t\}_t$. The ℓ_2 diameter of a set $\mathcal{X}$ is denoted by $D_{\mathcal{X}}$.

Outline of Paper. Sec. 2 reviews the related work about O-RANs and online learning. Sec. 3 presents the system model, the load assignment learning problem and its saddle-point reformulation. Sec. 4 provides a brief background on FTRL and introduces the optimistic FTRL algorithm for the load assignment problem; while Sec. 5 presents the model and algorithm for the TB threshold policy. We provide motivating experiments, and extensive simulation and experimental evaluation of the two algorithms in Sec. 6. The remaining proofs and additional results can be found in the Appendix.

2 LITERATURE REVIEW

2.1 Resource Management in vRANs

Resource management solutions for mobile networks can be broadly classified into those using analytical functions that map control actions to performance metrics, e.g., [21, 45, 77, 86]; solutions that employ offline-trained ML models, e.g., [22, 74]; and techniques that adapt to network conditions and user demands [4, 13, 39, 89, 91]. Unfortunately, function-based models rely on parameters that are most often unknown in vRANs; while the efficacy of ML models depends on the availability of *representative* training data [90]. Examples of more adaptive solutions include Bayesian learning for optimizing video analytics [39] and BS energy costs [14, 15, 17]; and Reinforcement Learning for spectrum management and wireless scheduling [2, 91], among many others. These approaches have high overhead, e.g., require expensive matrix inversions, and provide optimality guarantees only under stationary conditions; therefore, they are typically employed for longer-term static resource control policies. Other practical solutions tailored to vRAN resource management include hybrid offline-trained and online-adapted vBS workload predictions [38], or regression models [40], so as to increase the utilization of the employed CPUs. These works do not provide optimality or fairness guarantees, and operate in real-time as opposed to the non-RT scale of our assignment policy; thus can be used concurrently.

Here, we rely instead on the theory of online convex optimization (OCO) which: *(i)* does not require access to performance/cost functions or system/user-related parameters; and *(ii)* offers guarantees under a wide range of scenarios, including adversarial ones [92]. The robustness of

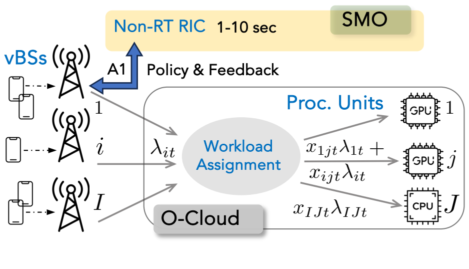
OCO is particularly useful for vRANs which, due to the virtualized vBS functions, exhibit volatile performance and, importantly, high energy costs [18, 26, 38, 40]. These experimental findings motivated the design of policies about the vBSs' transmission power, modulation and coding schemes, spectrum usage, and others, which aim to curb the vRANs energy costs [14, 15, 17]. Such policies can be devised centrally by the RIC and applied to different vBSs concurrently [33, 42]. These solutions become more interesting due to HAs that are increasingly common in industry-grade vBS-hosting platforms [32, 49]. HAs are already used in cloud computing where, due to their high cost, it is imperative to utilize them effectively [31, 54]. When it comes to vBS functions, the performance and energy costs of HAs are substantially different from CPUs as our experiments find, hence paving the road for new energy-saving policies. Our proposal includes a new near-RT *radio control* policy that decides the minimum TB size each user can employ; and a non-RT *compute-control* policy which assigns vBSs' workloads to CPU and HAs. Both policies optimize performance and energy consumption, while being fair in terms of the service offered to users and the energy costs dispersed across the PUs.

2.2 Fairness & Online Learning

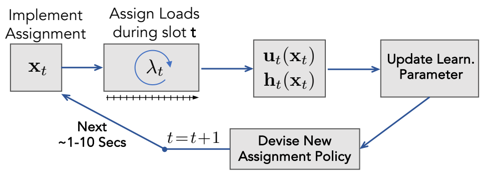
Fairness is a key metric in resource management and has been extensively applied in cloud computing [27, 87] and communication systems [5, 52], among many others [68, 71, 75]. More recently, [59, 60] focused on max-min throughput fairness in RANs, e.g., via spectrum management; [85] studied the fair allocation of computing capacity to vRAN functions and edge services; [67] considered cost-fairness in multi-tenant O-RANs where operators lease computing for their vBS functions; while [37, 45, 64] focus on virtualization and slicing. These interesting works, however, do not consider the inherent system and user dynamics in vRANs and/or do not provide fairness guarantees. Achieving fairness in such dynamic problems is indeed challenging, even from a theoretical point of view. Previous works have studied *slot fairness* criteria [50, 82, 83] where in each decision round the problem of fairness is tackled independently of the past or future decisions. By definition, the scope of these fairness metrics is limited and it results in higher price of fairness [25, 81]. More ambitious approaches attempt to achieve *horizon fairness*, where the fairness metric is enforced across the entire system operation, not instantly. We also refer the reader to the interesting work [6] that discusses fairness in wireless networks over multiple time scales.

Recent studies on horizon fairness assume the utility functions to be either known or non-adversarial, e.g., study the stochastic version, [6, 19, 23, 30, 44, 55, 81]. Instead, we target a framework that drops these assumptions. The closest to our work is [81], which we extend here in many ways. First, we use a novel *optimistic* learning fairness algorithm that leverages predictions for the performance and costs. Secondly, we consider a two-sided alpha-fairness criterion, i.e., w.r.t. cost savings across the servers and w.r.t. performance across users, where the two fairness parameters can be even different. And, finally, we employ a tailored learning algorithm with minimal computation and memory requirements. Namely, we rely on the FTRL framework [58] and draw ideas from optimistic learning [66, 76] that has been recently used, e.g., for caching [61] and network control [9]. Here, we extend the optimistic learning algorithms to our problems, aiming for low computation and memory requirements and constant (not only sublinear) regret for perfect predictions.

Finally, it is worth noting the connection of fairness with load-balancing techniques. The majority of studies in this latter area focus on the asymptotic regime and consider stochastic loads or servers with fixed capacities; see discussion of literature in [88]. Works that do drop these assumptions include [65] which assigns equal-length jobs with known deadlines; and [29] which considers max-min fairness from the servers' (minimize maximum load) or jobs' (maximize minimum service)



(a)


Implement Assignment
Assign Loads during slot t
$\mathbf{x}_t$
λ_t
$\mathbf{u}_t(\mathbf{x}_t)$
$\mathbf{h}_t(\mathbf{x}_t)$
Update Learn. Parameter
Next ~1-10 Secs
$t=t+1$
Devise New Assignment Policy

(b)

Fig. 3. **(a):** A non-RT controller at the Service & Management Orchestration (SMO) framework devises the load assignment policy every $\sim 1-10$ seconds and sends it to the vBSs via the A1 interface. **(b)**: Timing diagram of assignment implementation and learning policy.

perspective. Our model and motivation is different. We aim at fairness (i.e., balance) w.r.t. time-varying and unknown job-utility and server-cost functions, we make no assumptions about the load arrivals, and our policies operate at time-scales where queue stability is of no concern.

3 SYSTEM MODEL AND FAIRNESS REGRET

We start with the system model for the assignment problem; define the learning problem and regret metrics; and propose a saddle-point reformulation which is used in the algorithm design in Sec. 4.

3.1 Model & Problem Statement

We consider a vRAN with a set $\mathcal{I}$ of $I = |\mathcal{I}|$ vBSs, and a set $\mathcal{J}$ of $J = |\mathcal{J}|$ of PUs (or, servers) that comprise the O-Cloud. The operation of the system is time-slotted and the slot duration is considered $\sim (1 - 10)$ seconds, since this is a non-RT policy implemented by a RIC at the SMO, Fig. 3(a). We study the system for a set $\mathcal{T}$ of $T = |\mathcal{T}|$ slots, and focus on the more computation-demanding uplink [14, 18]. During every slot t, each vBS $i \in \mathcal{I}$ injects into the O-Cloud an amount of $\lambda_{it} \geq 0$ data (bytes), stemming from its users, and we define $\boldsymbol{\lambda}_t = (\lambda_{it}, i \in \mathcal{I})$. The required computations for these data, e.g., for FFT or FEC decoding, depend on their volume and wireless conditions that affect their SNR, see Fig. 1-2 and [18, 40]. Hence, in practice the value of $\boldsymbol{\lambda}_t$ and their computations are revealed at the end of each slot t. On the other hand, each server $j \in \mathcal{J}$ has computing capacity of C_{jt} cycles during each slot t, and we define $\boldsymbol{C}_t = (C_{jt}, j \in \mathcal{J})$. We study the general case where the capacities might change over time. Similar to the loads, we assume $\boldsymbol{C}_t$ becomes known at the end of each slot.

A non-RT controller decides the O-Cloud *assignment policy*, i.e., how much data (or, load) from each vBS will be routed to each server. We denote with $x_{ijt} \in [0, 1]$ the load portion of vBS i that is sent to server j during slot t, and hence $x_{ijt}\lambda_{it}$ is the assigned data from that vBS. We also define the assignment vector $\boldsymbol{x}_{it} = (x_{ijt}, \forall j \in \mathcal{J})$ for each vBS $i \in \mathcal{I}$; the vector $\boldsymbol{x}_{jt} = (x_{ijt}, \forall i \in \mathcal{I})$ for each server $j \in \mathcal{J}$; and the total assignment $\boldsymbol{x}_t = (x_{ijt}, \forall i \in \mathcal{I}, j \in \mathcal{J})$. These decisions are subject to a simplex constraint for each vBS, thus each $\boldsymbol{x}_t, t \in \mathcal{T}$, belongs to set:

$$\mathcal{X} = \left\{ \boldsymbol{x} \in [0, 1]^{I \cdot J} \ : \ \sum_{j \in \mathcal{J}} x_{ij} = 1, \forall i \in \mathcal{I} \right\}. \tag{1}$$

The assignment policy is updated at the beginning of each slot t and shapes the system performance during that slot. If the controller assigns more load to a server than its capacity, then (part of) this data will not be processed before its deadline [26]. This means that the associated vBSs will suffer reduced throughput [40, 41]. Thus, the benefit for a vBS when using a server decreases sharply when the total load approaches the server's capacity. We model this effect through a (possibly) time-varying utility vector function $\boldsymbol{u}_t(\boldsymbol{x}) = \big(u_{it}(\boldsymbol{x}), i \in \mathcal{I}\big)$, where $\boldsymbol{u}_t : \mathbb{R}^{I\times J} \mapsto \mathbb{R}_+^I$ is assumed non-negative and concave. Each element $u_{it}(\boldsymbol{x}) \in [u_{min}, u_{max}]$ denotes the performance for vBS $i \in \mathcal{I}$ under assignment $\boldsymbol{x}$, and captures the server heterogeneity, e.g., through $\boldsymbol{C}_t$.

Accordingly, we use functions $\boldsymbol{h}_t : \mathbb{R}^{I\times J} \mapsto \mathbb{R}_+^J$, to model the *energy cost savings* for the servers at each slot $t \in \mathcal{T}$, where $h_{jt}(\boldsymbol{x}) \in [h_{min}, h_{max}]$ is the cost reduction[5] of server $j \in \mathcal{J}$ under assignment $\boldsymbol{x}$. This reduction is calculated with reference to the (unknown) energy cost the server would have paid, had it served the entire load in the network. Put it differently, these functions model the benefits from dispersing the load across multiple servers instead of using only one. In line with prior works, e.g., [85], and based on our measurements (Sec. 6) we consider these functions to be non-negative and concave on $\boldsymbol{x}$. Our analysis can be applied to any type of utility and cost functions satisfying these minimal requirements, and we study a specific example in Sec. 6.

The goal of the controller is to devise a sequence of assignment policies $\{\boldsymbol{x}\}_{t=1}^T$ so as to achieve a two-sided fairness criterion: *(i)* fairness w.r.t. the average utility perceived by the vBSs over the horizon $\mathcal{T}$, i.e., w.r.t. $(1/T)\sum_{t\in\mathcal{T}} \boldsymbol{u}_t(\boldsymbol{x}_t)$; and *(ii)* fairness w.r.t. to the average energy cost savings of servers, i.e., $(1/T)\sum_{t\in\mathcal{T}} \boldsymbol{h}_t(\boldsymbol{x}_t)$. To do so, the controller needs to overcome two challenges. First, to decide the per-slot assignment $\boldsymbol{x}_t$ in a way that optimizes the immediate performance and costs while tracking these two long-term (*horizon*) fairness criteria. Secondly, it needs to achieve this balance without information about the system parameters $\{\boldsymbol{\lambda}_t, \boldsymbol{C}_t\}_t$, and functions $\{\boldsymbol{u}_t, \boldsymbol{h}_t\}_t$, which are time-varying, unknown, and revealed after each $\boldsymbol{x}_t$ is decided, see Fig. 3(b). In fact, we adopt the most general perturbation model where these parameters are assumed to be decided dynamically by an *adversary* aiming to deteriorate the system operation [92]. Clearly, a policy that performs well under these conditions, can also perform under more benign static or stationary scenarios[6]. For the fairness criteria, we employ the *generalized* α-fairness function [5, 63]:

$$F_\alpha(\boldsymbol{u}) \doteq \sum_{i\in\mathcal{I}} f_\alpha(u_i) \quad \text{where} \quad f_\alpha(u_i) \doteq \begin{cases} \frac{u_i^{1-\alpha}-1}{1-\alpha}, & \text{for } \alpha \in \mathbb{R}_{\geq 0}\backslash\{1\}, \\ \log(u_i), & \text{for } \alpha = 1. \end{cases} \tag{2}$$

Parameter α determines the type of fairness we wish to enforce; e.g., $\alpha=1$ yields the proportional fairness metric, while $\alpha \longrightarrow \infty$ leads to max-min fairness. We define a similar fairness function for the cost savings, $F_\beta(\boldsymbol{h}) = \sum_{j\in\mathcal{J}} f_\beta(h_j)$, where in general it can be $\alpha \neq \beta$.

We evaluate the efficacy of the assignment policies using the metric of *static regret* which is extended here to capture the two-sided horizon fairness as follows:

$$\begin{aligned} \mathcal{R}_T(F_\alpha, F_\beta) \doteq \sup_{\{\boldsymbol{u}_t,\boldsymbol{h}_t\}_{t=1}^T} \Bigg\{ & F_\alpha\left(\frac{1}{T}\sum_{t\in\mathcal{T}} \boldsymbol{u}_t(\boldsymbol{x}^\star)\right) + F_\beta\left(\frac{1}{T}\sum_{t\in\mathcal{T}} \boldsymbol{h}_t(\boldsymbol{x}^\star)\right) \\ & - F_\alpha\left(\frac{1}{T}\sum_{t\in\mathcal{T}} \boldsymbol{u}_t(\boldsymbol{x}_t)\right) - F_\beta\left(\frac{1}{T}\sum_{t\in\mathcal{T}} \boldsymbol{h}_t(\boldsymbol{x}_t)\right)\Bigg\}. \end{aligned} \tag{3}$$

This metric evaluates the policy that decides $\{\boldsymbol{x}_t\}$ dynamically over $\mathcal{T}$, by using a hypothetical benchmark $\boldsymbol{x}^\star$ that could be only devised with access at $t=0$ to all loads, capacities, and functions:

$$\boldsymbol{x}^\star = \arg\max_{x\in\mathcal{X}} \left\{ F_\alpha\left(\frac{1}{T}\sum_{t\in\mathcal{T}} \boldsymbol{u}_t(\boldsymbol{x})\right) + F_\beta\left(\frac{1}{T}\sum_{t\in\mathcal{T}} \boldsymbol{h}_t(\boldsymbol{x})\right)\right\}.$$

[5]Parameters h_{min} and h_{max}, as well as u_{min} and u_{max}, can be determined based on the vBS operation envelope.
[6]A policy designed for adversarial environments might not be, in general, ideal (i.e., best-performing) for static environments and can be outperformed by algorithms tailored for such specific scenarios, when one has guarantees for their existence.

We aim to find $\{\boldsymbol{x}_t\}_{t=1}^T$ that ensures the loss compared to $\boldsymbol{x}^\star$ will diminish to zero, for *any realization* of the unknown parameters, as is evidenced from the regret definition in (3).

3.2 Reformulation & Solution Approach

Unfortunately, off-the-shelf (online) convex optimization algorithms cannot be applied directly on this problem, due to the time-averaging in the argument of functions $F_\alpha(\cdot)$ and $F_\beta(\cdot)$, which does not allow the necessary (for these techniques) decomposition over time; see also [1, 81]. To tackle this issue, we introduce a *proxy function* $\Psi_t : \Theta \times \Phi \times \mathcal{X} \mapsto \mathbb{R}$ with two types of dual variables, $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{\phi} \in \Phi$, as follows:

$$\Psi_t(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}) = \Psi_t^\alpha(\boldsymbol{\theta}, \boldsymbol{x}) + \Psi_t^\beta(\boldsymbol{\phi}, \boldsymbol{x}), \tag{4}$$

where functions $\Psi_t^\alpha : \Theta \times \mathcal{X} \mapsto \mathbb{R}$ and $\Psi_t^\beta : \Phi \times \mathcal{X} \mapsto \mathbb{R}$, are defined as:

$$\Psi_t^\alpha(\boldsymbol{\theta}, \boldsymbol{x}) = (-F_\alpha)^\star(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \boldsymbol{u}_t(\boldsymbol{x}), \quad \Psi_t^\beta(\boldsymbol{\phi}, \boldsymbol{x}) = (-F_\beta)^\star(\boldsymbol{\phi}) - \boldsymbol{\phi}^\top \boldsymbol{h}_t(\boldsymbol{x}), \tag{5}$$

and the dual variables are bounded in $\Theta \doteq [-1/u^\alpha_{min}, -1/u^\alpha_{max}]^I$ and $\Phi \doteq [-1/h^\beta_{min}, -1/h^\beta_{max}]^J$. Function $(-F_\alpha)^\star(\cdot)$ is the Fenchel convex conjugate of $-F_\alpha\big(\boldsymbol{u}_t(\boldsymbol{x})\big)$ [20, Ch. 4], i.e.,

$$(-F_\alpha)^\star(\boldsymbol{\theta}) = \max_{\boldsymbol{x}\in\mathcal{X}} \left\{\boldsymbol{\theta}^\top \boldsymbol{u}_t(\boldsymbol{x}) - \big(-F_\alpha(\boldsymbol{u}_t(\boldsymbol{x}))\big)\right\} = \max_{\boldsymbol{x}\in\mathcal{X}} \left\{\boldsymbol{\theta}^\top \boldsymbol{u}_t(\boldsymbol{x}) + F_\alpha\big(\boldsymbol{u}(\boldsymbol{x})\big)\right\}, \tag{6}$$

and similarly we define $(-F_\beta)^\star(\boldsymbol{\phi})$ for function $-F_\beta\big(\boldsymbol{u}_t(\boldsymbol{x})\big)$. Interestingly, given the function structure in (2), these proxy functions can be expressed analytically as

$$\Psi_t^\alpha(\boldsymbol{\theta}, \boldsymbol{x}) = \sum_{i=1}^I \frac{\alpha(-\theta_i)^{1-1/\alpha} - 1}{1-\alpha} - \boldsymbol{\theta}^\top \boldsymbol{u}_t(\boldsymbol{x}), \quad \Psi_t^\beta(\boldsymbol{\phi}, \boldsymbol{x}) = \sum_{j=1}^J \frac{\beta(-\phi_j)^{1-1/\beta} - 1}{1-\beta} - \boldsymbol{\phi}^\top \boldsymbol{h}_t(\boldsymbol{x}), \tag{7}$$

and when $\alpha = 1$ we get $\Psi_t^\alpha(\boldsymbol{\theta}, \boldsymbol{x}) = -1 - \log(-\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \boldsymbol{u}_t(\boldsymbol{x})$, and similarly for $\beta = 1$ and Ψ_t^β.

These functions are suitable for our problem as we can recover the fairness objective with a minimization operation [20, Th. 4.8]. That is, leveraging their biconjugate equivalence we can write:

$$F_\alpha\big(\boldsymbol{u}_t(\boldsymbol{x})\big) = \min_{\boldsymbol{\theta}\in\Theta} \Psi_t^\alpha(\boldsymbol{\theta}, \boldsymbol{x}) \quad \text{and} \quad F_\beta\big(\boldsymbol{h}_t(\boldsymbol{x})\big) = \min_{\boldsymbol{\phi}\in\Phi} \Psi_t^\beta(\boldsymbol{\phi}, \boldsymbol{x}). \tag{8}$$

At the same time, $\Psi_t(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x})$ is linear on the utility and cost values, hence with this transformation we can maximize a (separable) sum of functions instead of a (non-separable) concave function of them. Putting these together, the problem we aim to solve has at its core the (per-slot) program:

$$\max_{x\in\mathcal{X}} \left\{F_\alpha\big(\boldsymbol{u}_t(\boldsymbol{x}_t)\big) + F_\beta\big(\boldsymbol{h}_t(\boldsymbol{x}_t)\big)\right\} = \max_{x\in\mathcal{X}} \left\{\min_{\theta\in\Theta} \Psi_t^\alpha(\boldsymbol{\theta}, \boldsymbol{x}) + \min_{\boldsymbol{\phi}\in\Phi} \Psi_t^\beta(\boldsymbol{\phi}, \boldsymbol{x})\right\}, \tag{9}$$

which we will tackle with a saddle-point algorithm that updates the primal and dual variables successively, performing independent (but coordinated) learning in the primal and dual space. In particular, we will be running an OCO algorithm on $\boldsymbol{x}$ to bound the primal-space regret:

$$\mathcal{R}_T^x \doteq \sum_{t=1}^T \Big(\Psi_t(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t, \boldsymbol{x}) - \Psi_t(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t, \boldsymbol{x}_t)\Big), \ \forall \boldsymbol{x} \in \mathcal{X}, \tag{10}$$

and similarly, we will learn using the proxy function in the dual spaces, to bound:

$$\mathcal{R}_T^\theta \doteq \sum_{t=1}^T \Big(\Psi_t^\alpha(\boldsymbol{\theta}_t, \boldsymbol{x}_t) - \Psi_t^\alpha(\boldsymbol{\theta}, \boldsymbol{x}_t)\Big), \forall \boldsymbol{\theta} \in \Theta, \quad \mathcal{R}_T^\phi \doteq \sum_{t=1}^T \Big(\Psi_t^\beta(\boldsymbol{\phi}_t, \boldsymbol{x}_t) - \Psi_t^\beta(\boldsymbol{\phi}, \boldsymbol{x}_t)\Big), \ \forall \boldsymbol{\phi} \in \Phi. \tag{11}$$

We will show in the next section that we can use these regret bounds to upper-bound the horizon-fair regret $\mathcal{R}_T(F_\alpha, F_\beta)$, which is the goal of the RIC here.

4 LEARNING ALGORITHMS AND ASSIGNMENT POLICY

We now present the learning algorithms in the primal and dual space and characterize their regret bounds, which we then combine to build the controller's assignment policy and assess its regret.

4.1 OFTRL Algorithms

We will perform the RIC policy learning using *optimistic FTRL* algorithms [66, 76]. In the OFTRL template, the variables are updated at the beginning of each new slot $t+1$ using a time-varying regularizer $r_{1:t}(\boldsymbol{x})$, all past function gradients and a prediction for the function gradient at slot $t+1$ (the optimistic element). Such predictions, if incorporated carefully, can improve the learning rate when accurate, without sacrificing performance when they are inaccurate. For example, one can use past values of vBS loads as a prediction for the next loads without risking no-learning conditions if there is a distribution shift. Optimistic learning has been recently used, e.g., for content caching [61] or routing [9], but not for vRANs and not in conjunction with allocative fairness.

The performance of such FTRL algorithms is shaped by the regularizers. A typical choice is the quadratic regularizer $r_{1:t}(\boldsymbol{x})=\frac{\sigma_{1:t}}{2}\|\boldsymbol{x}\|^2$ and its proximal variant which uses instead $\|\boldsymbol{x}-\boldsymbol{x}_t\|^2$. Parameters $\{\sigma_t\}$ encode information about the system properties and the predictions' accuracy. When the constraint is a simplex, the entropic regularizer $r_{1:t}(\boldsymbol{x})=\sigma_t\sum_{i\in\mathcal{I}}x_i\log(x_i)$ allows a closed-form derivation of $\boldsymbol{x}_t$ and achieves lower dependency on the decision space diameter. Proximal regularizers require more memory and computations to optimize the variables, but achieve $\mathcal{O}(1)$ regret when all predictions are accurate. On the other hand, non-proximal regularizers (as the entropic) are computationally-efficient but yield sublinear (not constant) regret $\mathcal{O}(\sqrt{T})$ even with perfect predictions, see [66]. Here, we use a quadratic regularizer in the dual space and an entropic one for the primal space. What is more, we tune these algorithms to achieve $\mathcal{O}(1)$ regret for perfect predictions (despite being non-proximal), and provide closed-form derivations in both cases.

4.1.1 *OFTRL with Quadratic Regularizer.*

We start with the analysis of the learning in the dual spaces Θ and Φ. The proposed OFTRL dual updates for these minimization problems are:

$$\boldsymbol{\theta}_{t+1}=\arg\min_{\boldsymbol{\theta}\in\Theta}\left\{q_{1:t}(\boldsymbol{\theta})+\boldsymbol{\theta}^\top(\boldsymbol{\kappa}_{1:t}+\tilde{\boldsymbol{\kappa}}_{t+1})\right\}, \tag{12}$$

$$\boldsymbol{\phi}_{t+1}=\arg\min_{\boldsymbol{\phi}\in\Phi}\left\{p_{1:t}(\boldsymbol{\phi})+\boldsymbol{\phi}^\top(\boldsymbol{\mu}_{1:t}+\tilde{\boldsymbol{\mu}}_{t+1})\right\}, \tag{13}$$

where $q_{1:t}(\boldsymbol{\theta})=\sum_{\tau=0}^{t}q_\tau(\boldsymbol{\theta})$ and $p_{1:t}(\boldsymbol{\phi})=\sum_{\tau=0}^{t}p_\tau(\boldsymbol{\phi})$ are the aggregate dual regularizing functions imposed at slot t; vectors $\boldsymbol{\kappa}_{1:t}=\sum_{\tau=1}^{t}\nabla_{\boldsymbol{\theta}}\Psi_\tau^{a}(\boldsymbol{\theta}_\tau,\boldsymbol{x}_\tau)$, $\boldsymbol{\mu}_{1:t}=\sum_{\tau=1}^{t}\nabla_{\boldsymbol{\phi}}\Psi_\tau^{\beta}(\boldsymbol{\phi}_\tau,\boldsymbol{x}_\tau)$ are the aggregate dual gradients; and $\tilde{\boldsymbol{\kappa}}_{t+1}$, $\tilde{\boldsymbol{\mu}}_{t+1}$ denote the respective gradient predictions for $t+1$. Following the rationale in [62, 66] and based on the geometry of these spaces, we propose the regularizers:

$$q_{1:t}(\boldsymbol{\theta})=\frac{\sigma_{1:t}}{2}\|\boldsymbol{\theta}\|_2^2 \quad\text{where}\quad \sigma_{1:t}=\sigma\sqrt{\sum_{\tau=1}^{t}\|\boldsymbol{\kappa}_\tau-\tilde{\boldsymbol{\kappa}}_\tau\|_2^2},\quad \sigma=2\sqrt{2}/D_\Theta, \tag{14}$$

$$p_{1:t}(\boldsymbol{\phi})=\frac{\xi_{1:t}}{2}\|\boldsymbol{\phi}\|_2^2 \quad\text{where}\quad \xi_{1:t}=\xi\sqrt{\sum_{\tau=1}^{t}\|\boldsymbol{\mu}_\tau-\tilde{\boldsymbol{\mu}}_\tau\|_2^2},\quad \xi=2\sqrt{2}/D_\Phi, \tag{15}$$

which impose regularization commensurate to the prediction errors up to each slot $t\in\mathcal{T}$. It follows that $q_{1:t}(\boldsymbol{\theta})$ is 1-strongly-convex w.r.t. the norm $\|\boldsymbol{\theta}\|_{(t)}=\sqrt{\sigma_{1:t}}\|\boldsymbol{\theta}\|$, which has dual norm $\|\boldsymbol{\theta}\|_{(t),\star}=\|\boldsymbol{\theta}\|/\sqrt{\sigma_{1:t}}$, and similarly for $p_{1:t}(\boldsymbol{\phi})$, see [58].

If we apply the OFTRL updates (12)-(13) with regularizers (14)-(15), we can upper-bound the regret in the dual spaces as the next result states, which holds as is for Φ as well.

Lemma 4.1. *For a compact convex set* Θ*, update* (12) *with regularizer* (14) *yields regret:*

$$\mathcal{R}_T^{\theta} \le 4\sqrt{2}D_{\Theta}\sqrt{\sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}. \tag{16}$$

This result improves the optimistic regret bound of quadratic regularizers by enabling constant regret $O(1)$, as opposed to sublinear (not constant) regret [66], in the case of perfect predictions.

4.1.2 OFTRL Algorithm with Entropic Regularizer. For the primal update we employ entropic regularization due to the multi-simplex structure of $\mathcal{X}$. The update for this problem is[7]:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ r_{1:t}(\boldsymbol{x}) - x^{\top}\left(\boldsymbol{g}_{1:t} + \boldsymbol{w}_{1:t} + \tilde{\boldsymbol{g}}_{t+1} + \tilde{\boldsymbol{w}}_{t+1}\right)\right\} \tag{17}$$

where $r_{1:t}(\boldsymbol{x})$ is the aggregate regularization at t; vectors $\boldsymbol{g}_{1:t} = \sum_{\tau=1}^{t} \nabla_{\boldsymbol{x}} \Psi_{\tau}^{a}(\boldsymbol{\theta}_{\tau}, \boldsymbol{x}_{\tau})$ and $\boldsymbol{w}_{1:t} = \sum_{\tau=1}^{t} \nabla_{\boldsymbol{x}} \Psi_{\tau}^{\beta}(\boldsymbol{\phi}_{\tau}, \boldsymbol{x}_{\tau})$ are the aggregate primal-space gradients; and $\tilde{\boldsymbol{g}}_{t+1}$ and $\tilde{\boldsymbol{w}}_{t+1}$ the gradient predictions for $t+1$. The proposed entropic regularizer for this multi-simplex constraint is:

$$r_{1:t}(\boldsymbol{x}) = \frac{\eta_{1:t}}{2}\left(I \log J + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} x_{ij}\log x_{ij}\right), \quad \text{where} \quad \eta_{1:t} = \eta\sqrt{\sum_{\tau=1}^{t} \|\boldsymbol{g}_{\tau} + \boldsymbol{w}_{\tau} - \tilde{\boldsymbol{g}}_{\tau} - \tilde{\boldsymbol{w}}_{\tau}\|_{\infty}^2}. \tag{18}$$

Each $r_{1:t}(\boldsymbol{x})$ is now 1-strongly-convex w.r.t. norm $\|\boldsymbol{x}\|_{(t)} = \|\boldsymbol{x}\|_1 \left(\eta_{1:t}/I\right)^{1/2}$ (see Lemma 8.1 in Appendix). This update yields regret in the primal space that is upper bounded by the next Lemma.

Lemma 4.2. *For the convex set* $\mathcal{X}$ *defined in* (1)*, the update* (17) *with regularizer* (18) *ensures:*

$$\mathcal{R}_T^{x} \le \left(\frac{\sqrt{2}I}{\eta} + \frac{\eta I \log J}{2}\right)\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_{\infty}^2} \quad \textit{where} \quad \eta = \min\left\{\frac{1}{2}, \sqrt{\frac{2\sqrt{2}}{\log J}}\right\}. \tag{19}$$

Similarly to $\mathcal{R}_T^{\theta}$ and $\mathcal{R}_T^{\phi}$, this result ensure regret $\mathcal{R}_T^{x} = O(1)$ when the predictions are perfect, while we still get $\mathcal{R}_T^{x} = O(\sqrt{T})$ even when the predictions are maximally inaccurate.

4.1.3 Implementation. Given the tight deadlines of the vBS functions and the scale of vRANs, it is imperative the algorithm to be lightweight. To that end, we solve analytically its core optimization steps. Applying first-order optimality conditions on (7), we can express the partial gradients as:

$$\boldsymbol{g}_t = \left(-\sum_{i=1}^{I} \theta_{it}\frac{\partial u_{it}(\boldsymbol{x}_t)}{\partial x_{ijt}},\ i\in\mathcal{I}, j\in\mathcal{J}\right), \quad \boldsymbol{w}_t = \left(-\sum_{j=1}^{J} \phi_{jt}\frac{\partial h_{ht}(\boldsymbol{x}_t)}{\partial x_{ijt}},\ i\in\mathcal{I}, j\in\mathcal{J}\right), \quad \text{and} \tag{20}$$

$$\boldsymbol{\kappa}_t = \left(-(-\theta_{it})^{-\frac{1}{\alpha}} - u_{it}(\boldsymbol{x}_t),\ i\in\mathcal{I}\right), \qquad \boldsymbol{\mu}_t = \left(-(-\phi_{jt})^{-\frac{1}{\beta}} - h_{jt}(\boldsymbol{x}_t),\ j\in\mathcal{J}\right). \tag{21}$$

Furthermore, both the dual and primal variable updates can be performed with closed-form expressions leveraging the following formulas.

Proposition 1. The closed-form solution to $\boldsymbol{\theta}_{t+1}$ in iteration (12) is given by

$$\theta_{i,t+1} = \min\left\{\max\left\{-\frac{\kappa_{i,1:t} + \tilde{\kappa}_{i,t+1}}{\frac{2\sqrt{2}}{D_{\Theta}}\sqrt{\sum_{\tau=1}^{t}\|\boldsymbol{\kappa}_{\tau} - \tilde{\boldsymbol{\kappa}}_{\tau}\|^2}}, \frac{-1}{u_{min}^{\alpha}}\right\}, \frac{-1}{u_{max}^{\alpha}}\right\}, \forall i\in\mathcal{I}. \tag{22}$$

A similar expression can be derived for variables $\{\phi_{jt}\}$, while for the primal update we can use:

[7]Note that, as the primal-space problem is a minimization one the aggregate gradient here has a minus sign.

Proposition 2. The closed-form solution to $\boldsymbol{x}_t$ in iteration (17) is given by:

$$x_{ij,t+1} = \frac{\exp\left(2\omega_{ijt}/\eta_{1:t}\right)}{\sum_{j\in\mathcal{J}} \exp\left(2\omega_{ijt}/\eta_{1:t}\right)}, \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{23}$$

where $\omega_{ijt} \doteq g_{ij,1:t} + w_{ij,1:t} + \tilde{g}_{ij,t+1} + \tilde{w}_{ij,t+1}, \forall i, j, t$, and $\eta_{1:t}$ is given by (18).

Such closed-form expressions are commonly used for entropic regularizers over *one* simplex [79], and we extend this idea for the *multi-simplex* set $\mathcal{X}$. This allows to run the primal updates with $O(1)$ memory since we maintain only the aggregate gradients, and with $O(1)$ computation time.

4.2 Horizon-Fair Assignment Policy

We leverage the above results of regret and the expressions for the primal and dual updates to design the optimistic FTRL policy for the assignment problem; see Algorithm 1. The initialization (lines 1-2) requires minimal information, i.e., the dimension of the primal and dual-space constraint sets and the number of vBSs and servers. The first assignment is drawn randomly (line 3). After running the first slot with policy $\boldsymbol{x}_1$, we observe the utility and cost functions and the respective gradients (line 5), as these have been set by the adversary. Accordingly we calculate the primal and dual gradients (lines 6-7) using the provided closed-form expressions, and we obtain the predicted gradient vectors for the next slot (line 8). This information is used to build the primal and dual regularizers and calculate the assignment policy $\boldsymbol{x}_{t+1}$ and the dual variables, $\boldsymbol{\theta}_{t+1}$ and $\boldsymbol{\phi}_{t+1}$, that will be used during the next slot (line 9). These steps are repeated throughout the horizon $\mathcal{T}$, which is not required as input to the algorithm nor has to be fixed in advance. The performance of Algorithm 1 is characterized by the following theorem.

Theorem 4.3. *Algorithm 1 attains regret:*

$$\mathcal{R}_T(F_\alpha, F_\beta) \le \frac{1}{T}\left(\frac{\sqrt{2}I}{\eta} + \frac{\eta I \log J}{2}\right)\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2} + \frac{4\sqrt{2}D_\Theta}{T}\sqrt{\sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}$$
$$+ \frac{4\sqrt{2}D_\Phi}{T}\sqrt{\sum_{t=1}^{T} \|\boldsymbol{\mu}_t - \tilde{\boldsymbol{\mu}}_t\|_2^2} + \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_T\right)^\top \boldsymbol{u}_t(\boldsymbol{x}^\star) + \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{\phi}_t - \bar{\boldsymbol{\phi}}_T\right)^\top \boldsymbol{h}_t(\boldsymbol{x}^\star)$$

where $D_\Theta = (\frac{1}{u^\alpha_{min}} - \frac{1}{u^\alpha_{max}})\sqrt{I}$, $D_\Phi = (\frac{1}{h^\beta_{min}} - \frac{1}{h^\beta_{max}})\sqrt{J}$, $\bar{\boldsymbol{\theta}}_T \doteq \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_t$, $\bar{\boldsymbol{\phi}}_T \doteq \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\phi}_t$.

Proof. First, we observe that since we perform OFTRL on the dual variables $\boldsymbol{\theta}$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\Psi_t^a(\boldsymbol{\theta}_t, \boldsymbol{x}_t) - \frac{1}{T}\sum_{t=1}^{T}\Psi_t^a(\boldsymbol{\theta}, \boldsymbol{x}_t) \le \frac{R_T^\theta}{T} = \frac{4\sqrt{2}D_\Theta}{T}\sqrt{\sum_{t=1}^{T}\|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}.$$

where the last step follows from Lemma 4.1. Similarly, for the dual variables $\boldsymbol{\phi}$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\Psi_t^\beta(\boldsymbol{\phi}_t, \boldsymbol{x}_t) - \frac{1}{T}\sum_{t=1}^{T}\Psi_t^\beta(\boldsymbol{\phi}, \boldsymbol{x}_t) \le \frac{R_T^\phi}{T} = \frac{4\sqrt{2}D_\Phi}{T}\sqrt{\sum_{t=1}^{T}\|\boldsymbol{\mu}_t - \tilde{\boldsymbol{\mu}}_t\|_2^2}.$$

Algorithm 1 Fair and Balanced Assignment Policy (non-RT RIC)

```
Require: $I, J$, Multi-simplex $\mathcal{X} \in \mathbb{R}^{I\times J}$, $\alpha, \beta \geq 0$, $[u^{\alpha}_{min}, u^{\alpha}_{max}]$, $[h^{\beta}_{min}, h^{\beta}_{max}]$.
 1: $\Theta = \left[-1/u^{\alpha}_{min}, -1/u^{\alpha}_{max}\right]^I$, $\Phi = \left[-1/h^{\beta}_{min}, -1/h^{\beta}_{max}\right]^J$        ▷ Initialize the dual spaces
 2: $\eta = \min\left\{1/2, \sqrt{2\sqrt{2}/\log J}\right\}$, $\sigma = 2\sqrt{2}/D_{\Theta}$, $\xi = 2\sqrt{2}/D_{\Phi}$        ▷ Initialize the regul. parameters
 3: $\boldsymbol{x}_1 \in \mathcal{X}, \boldsymbol{\theta}_1 \in \Theta, \boldsymbol{\phi}_1 \in \Phi$        ▷ Initialize primal and dual vars
 4: for $t = 1$ to $T$ do
 5:     Observe $\boldsymbol{u}_t(\boldsymbol{x}_t), \boldsymbol{h}_t(\boldsymbol{x}_t), \nabla_{\boldsymbol{x}}\boldsymbol{u}_t(\boldsymbol{x}_t), \nabla_{\boldsymbol{x}}\boldsymbol{h}_t(\boldsymbol{x}_t)$        ▷ Adversary selects losses
 6:     Compute primal gradients $\boldsymbol{g}_t, \boldsymbol{w}_t$ with (20)
 7:     Compute dual gradients $\boldsymbol{\kappa}_t, \boldsymbol{\mu}_t$ with (21)
 8:     Obtain predictions $\tilde{\boldsymbol{g}}_{t+1}, \tilde{\boldsymbol{w}}_{t+1}, \tilde{\boldsymbol{k}}_{t+1}$ and $\tilde{\boldsymbol{\mu}}_{t+1}$
 9:     Compute $\boldsymbol{x}_{t+1}$ with (23), $\boldsymbol{\theta}_{t+1}$ and $\boldsymbol{\phi}_{t+1}$ with (22)        ▷ Update assignment and dual vars
10: end for
```

On the other hand, the OFTRL on the primal variables $\boldsymbol{x}$ with the above regularizer, yields:

$$\frac{1}{T}\sum_{t=1}^{T}\Psi_t^{a}(\boldsymbol{\theta}_t, \boldsymbol{x}^{\star}) + \frac{1}{T}\sum_{t=1}^{T}\Psi_t^{\beta}(\boldsymbol{\theta}_t, \boldsymbol{x}^{\star}) - \frac{1}{T}\sum_{t=1}^{T}\Psi_t^{a}(\boldsymbol{\theta}_t, \boldsymbol{x}_t) - \frac{1}{T}\sum_{t=1}^{T}\Psi_t^{\beta}(\boldsymbol{\theta}_t, \boldsymbol{x}_t) \leq$$

$$\frac{R_T^x}{T} = \left(\frac{\sqrt{2}I}{\eta T} + \frac{\eta I \log J}{2T}\right)\sqrt{\sum_{t=1}^{T}\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_{\infty}^2}, \tag{24}$$

where we applied the regret bound from Lemma 4.2. Now, using (24) we can write:

$$\frac{1}{T}\sum_{t=1}^{T}\Psi_t^{a}(\boldsymbol{\theta}_t, \boldsymbol{x}_t) + \frac{1}{T}\sum_{t=1}^{T}\Psi_t^{\beta}(\boldsymbol{\phi}_t, \boldsymbol{x}_t) + \mathcal{R}_T^x \geq \frac{1}{T}\sum_{t=1}^{T}\Psi_t^{a}(\boldsymbol{\theta}_t, \boldsymbol{x}^{\star}) + \frac{1}{T}\sum_{t=1}^{T}\Psi_t^{\beta}(\boldsymbol{\phi}_t, \boldsymbol{x}^{\star})$$

$$= \frac{1}{T}\left[\sum_{t=1}^{T}(-F_a)^{\star}(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t^{\top}\boldsymbol{u}_t(\boldsymbol{x}^{\star})\right] + \frac{1}{T}\left[\sum_{t=1}^{T}(-F_{\beta})^{\star}(\boldsymbol{\phi}_t) - \boldsymbol{\phi}_t^{\top}\boldsymbol{h}_t(\boldsymbol{x}^{\star})\right]$$

$$\geq (-F_a)^{\star}(\bar{\boldsymbol{\theta}}) - \bar{\boldsymbol{\theta}}^{\top}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{u}_t(\boldsymbol{x}^{\star})\right) - \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}})^{\top}\boldsymbol{u}_t(\boldsymbol{x}^{\star})$$

$$+ (-F_{\beta})^{\star}(\bar{\boldsymbol{\phi}}) - \bar{\boldsymbol{\phi}}^{\top}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{h}_t(\boldsymbol{x}^{\star})\right) - \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\phi}_t - \bar{\boldsymbol{\phi}})^{\top}\boldsymbol{h}_t(\boldsymbol{x}^{\star})$$

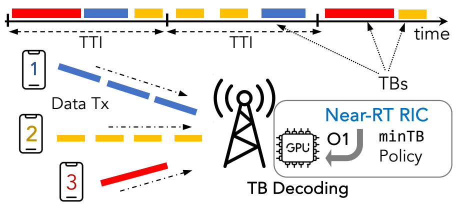
$$\geq \min_{\boldsymbol{\theta}\in\Theta}\left\{(-F_{\alpha})^{\star}(\boldsymbol{\theta}) - \boldsymbol{\theta}^{\top}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{u}_t(\boldsymbol{x}^{\star})\right)\right\} - \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}})^{\top}\boldsymbol{u}_t(\boldsymbol{x}^{\star})$$

$$+ \min_{\boldsymbol{\phi}\in\Phi}\left\{(-F_{\beta})^{\star}(\boldsymbol{\phi}) - \boldsymbol{\phi}^{\top}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{h}_t(\boldsymbol{x}^{\star})\right)\right\} - \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\phi}_t - \bar{\boldsymbol{\phi}})^{\top}\boldsymbol{h}_t(\boldsymbol{x}^{\star})$$

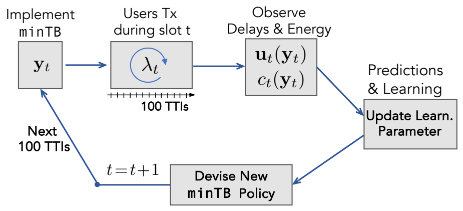
$$= F_a\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{u}_t(\boldsymbol{x}^{\star})\right) + F_{\beta}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{h}_t(\boldsymbol{x}^{\star})\right) - \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}})^{\top}\boldsymbol{u}_t(\boldsymbol{x}^{\star}) - \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\phi}_t - \bar{\boldsymbol{\phi}})^{\top}\boldsymbol{h}_t(\boldsymbol{x}^{\star})$$

We conclude by rearranging and using the biconjugate equivalence (8) for $\Psi_t^{a}(\boldsymbol{\theta}_t, \boldsymbol{x}_t)$, $\Psi_t^{\beta}(\boldsymbol{\phi}_t, \boldsymbol{x}_t)$. □

Discussion. There are some important notes in order here. First, observe the last two terms in the regret bound which quantify how much each dual vector deviates from its average (over $\mathcal{T}$).



(a)



(b)

Fig. 4. **(a):** The near-RT controller decides the TB threshold (`minTB`) policy for each user at each slot; and the TBs are processed at a HA-equiped server. **(b):** Timing diagram of the learning algorithm for the `minTB` policy.

These deviations depend on the type of the adversary, and remain sublinear under certain general conditions. Namely, the utility and cost functions can change in a non-i.i.d. fashion, even arbitrarily, as long as their perturbations remain within a sublinearly-growing perturbation budget. And there are two types of such budgets: budgeted severity, where we measure the severity of the adversary by summing the absolute value of (utility and cost) perturbations for the entire time-horizon; and partitioned severity, where we divide the time-horizon into contiguous partitions and calculate the absolute value of perturbations over each partition. As long as the perturbations satisfy at least one budget condition, the regret will remain sublinear. We refer the reader to [81] for further details, and stress that this condition is significantly milder than those in prior static or stochastic fairness frameworks [6, 43, 69]. We provide instances of such adversarial environments in Sec. 6.

The theorem also highlights the effect of predictions. The first two terms of the regret bound are eliminated when the predictions are perfect, while the algorithm suffers additional regret which is commensurate to the prediction errors (measured with the ℓ_∞ norm). In any case, these terms remain below $O(\sqrt{T})$. This reveals that predictions expedite the learning process while we retain the worst-case guarantees when they are inaccurate. Observe also that the bound depends on the numbers of servers only logarithmically, a known advantage of entropic regularizers, but has linear dependency on the number of vBSs. This is due to the structure of the constraint set $\mathcal{X}$ which consists of I (not 1) simplices. Similarly, the diameters D_Θ and D_Φ, which depend on the minimum and maximum utility and cost values, affect only linearly the regret bound.

Finally, regarding its implementation, leveraging the closed-form expressions for the decision updates, Algorithm 1 can be executed with $O(1)$ memory and $O(1)$ calculations, without the need to solve any optimization problem at runtime. At the same time, the algorithm is oblivious to user demands, system state (e.g., costs and available capacity), and channel conditions. These two features, along with its general convergence properties, make the proposed framework particularly useful from a practical point of view. As a last note, we wish to stress that our work advances the state-of-the-art by using closed-form expressions and predictions, and importantly by combining two different fairness metrics. An O-RAN operator will, of course, need to normalize carefully the utility and cost functions in order to achieve the desirable balance of these metrics, which is also affected by the values of α and β. For instance, one can divide each function with its maximum attainable value or simply scale them with a properly-selected parameter. We explore this aspect experimentally in Sec. 6.1.

5 FAIR SERVICE OF USERS AND VBS COST MINIMIZATION

Next, we study how a vBS can serve fairly its users in terms of latency by controlling the minimum size of their transmitted TBs, and minimize its own energy cost at the same time. Setting a threshold for the minimum TB size, the vBS prevents short TB transmissions that, as our experiments show

(Sec. 6.2) increase the energy cost. On the other hand, such thresholds introduce waiting times for users that might be non-negligible, e.g., for latency-critical services. According to O-RAN specifications and previous feasibility studies, e.g., [14–16], such radio control policies can be devised by a near-RT RIC and implemented with msec granularity and on per-user basis. We abuse slightly the notation here by redefining some parameters and variables.

5.1 Model

We consider a vBS that serves a set $\mathcal{I}$ of $I = |\mathcal{I}|$ users during a time period of T slots, where each slot consists of N TTIs (e.g., $N = 100$), and we focus on the uplink again. During each slot t, each user $i \in \mathcal{I}$ creates a certain amount of traffic (bytes) that needs to be transmitted to the vBS. We denote with $\mu_{itn} \geq 0$ the bytes created by user i from the beginning of the slot up to TTI n, and define the vectors $\boldsymbol{\mu}_{it} = (\mu_{itn}, n \leq N)$ for each user i and each slot t, and the vector $\boldsymbol{\mu}_t = (\boldsymbol{\mu}_{it}, i \in \mathcal{I})$ for the data of all users in slot t. The uplink transmission of a user is realized as soon as, and as many times as, its accumulated buffer load reaches the minimum TB size `minTB`. We denote with $\boldsymbol{y}_t = (y_{it} \geq 0, i \in \mathcal{I})$ the vector of `minTB` values for slot t , which in the general case can be different for each user. These values are upper-bounded by the total number K of transport blocks a vBS can support[8]. Hence, each $\boldsymbol{y}_t$ belongs to the set $\mathcal{Y} = [0, K]^I$.

The `minTB` strategy $\boldsymbol{y}_t$ is decided by the vBS at the beginning of each slot in order to balance the service latency and its energy cost when processing the transmitted data. Our experiments show that large TB values improve the energy consumption per processed bit (J/b); yet they induce longer waiting times for the user traffic, see Sec. 6. Clearly, the more data is required before an uplink transmission is initiated, the more the user needs to wait to receive service. We consider a general model where the utility function $u_{it} : \mathbb{R}^I \mapsto \mathbb{R}_+$ denotes the (expected) performance perceived by user i when the `minTB` strategy is $\boldsymbol{y}_t$. The vector $\boldsymbol{u}_t(\boldsymbol{y}) = (u_{it}(\boldsymbol{y}), i \in \mathcal{I})$ can measure directly the latency or a proxy metric such as time the user (MAC layer) buffer is empty[9] as in [16]. Furthermore, we denote with $c_t : \mathbb{R}^I \mapsto \mathbb{R}_+$ the vBS energy cost, which is considered to be convex and decreasing on $\boldsymbol{y}$. Our analysis below does not require any further assumptions on these utility and energy cost functions, while in Sec. 6 we provide examples based on testbed measurements.

The vBS aims to maximize the long-term latency fairness and minimize the average energy cost:

$$G_\alpha(\{\boldsymbol{y}_t\}_t) = F_\alpha\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{u}_t(\boldsymbol{y}_t)\right) - \frac{1}{T}\sum_{t=1}^{T} c_t(\boldsymbol{y}_t), \tag{25}$$

and to do so with a dynamic `minTB` policy $\{\boldsymbol{y}_t\}_t$ which ensures sublinear regret:

$$\mathcal{R}_T(G_\alpha) \doteq \sup_{\{\boldsymbol{u}_t, c_t\}_{t=1}^{T}} \left\{G_\alpha\left(\boldsymbol{y}^\star\right) - G_\alpha(\{\boldsymbol{y}_t\}_t)\right\} \tag{26}$$

where $G_\alpha\left(\boldsymbol{y}^\star\right)$ is the best performance (fairness and cost) that can be achieved if at $t = 0$ the utilities and costs for the entire $\mathcal{T}$ were known. This metric differs from the fairness-only criterion of the previous section due to the requirement for cost reduction and the constraints' geometry.

5.2 Algorithm & Regret Bounds

The algorithm for this problem is based on the following modified proxy function:

$$\Psi_t^c(\boldsymbol{\theta}_t, \boldsymbol{y}_t) \doteq (-F_\alpha)^\star(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t^\top \boldsymbol{u}_t(\boldsymbol{y}_t) - c_t(\boldsymbol{y}_t). \tag{27}$$

[8]Depending on the channel conditions, the actual number and size of transport blocks the vBS can support might fluctuate. Here, K is the maximum possible number, and at each slot the exact bound is set by the vBS real-time scheduler.
[9]Recall that buffer queue length minimization is commonly used for reducing network delay, see e.g., [70].

Algorithm 2 Fair and Cost-efficient `minTB` Policy

```
Require: Compact convex set $\mathcal{Y} \in \mathbb{R}^I$, $\alpha \geq 0$, $[u_{min}, u_{max}]$
1: $\Theta = \left[-1/u_{min}^{\alpha}, -1/u_{max}^{\alpha}\right]^I$                ▷ Initialize the dual space
2: $\sigma = 2\sqrt{2}/D_{\Theta}$, $\eta = 2\sqrt{2}/D_{\mathcal{Y}}$                ▷ Initialize the regul. parameters
3: $\boldsymbol{y}_1 \in \mathcal{Y}, \boldsymbol{\theta}_1 \in \Theta$                ▷ Initialize the primal and dual vars
4: for $t = 1$ to $T$ do
5:     Observe $\boldsymbol{u}_t(\boldsymbol{y}_t)$, $c_t(\boldsymbol{y}_t)$, $\nabla_{\boldsymbol{y}} \boldsymbol{u}_t(\boldsymbol{y}_t)$, $\nabla_{\boldsymbol{y}} c_t(\boldsymbol{y}_t)$                ▷ Incur reward and loss
6:     Compute gradients $\boldsymbol{s}_t$, $\boldsymbol{m}_t$.
7:     Obtain gradient predictions $\tilde{\boldsymbol{s}}_{t+1}$ and $\widetilde{\boldsymbol{m}}_{t+1}$
8:     Compute $\boldsymbol{y}_{t+1}$ using (28) and $\boldsymbol{\theta}_{t+1}$ using (29).
9: end for
```

The analysis is based on the observation that the addition of the cost function $c_t(\cdot)$, which is independent of the dual variables, does not affect the algebraic operations on the proxy function. The primal OFTRL update is:

$$\boldsymbol{y}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{Y}} \left\{ r_{1:t}(\boldsymbol{y}) - \boldsymbol{y}^\top \left(\boldsymbol{s}_{1:t} + \tilde{\boldsymbol{s}}_{t+1}\right) \right\}, \quad \text{with } r_{1:t}(\boldsymbol{y}) = \frac{\eta \|\boldsymbol{y}\|^2}{2} \sqrt{\sum_{\tau=1}^{t} \|\boldsymbol{s}_\tau - \tilde{\boldsymbol{s}}_\tau\|_2^2} \tag{28}$$

where $\boldsymbol{s}_t = \nabla_{\boldsymbol{y}} \Psi_t^c(\boldsymbol{\theta}_t, \boldsymbol{y}_t)$ is the gradient of the proxy function w.r.t. the primal variables in slot t, and includes both the utility and the cost function differential (a linear operation), and $\tilde{\boldsymbol{s}}_{t+1}$ is the respective utility and cost gradient prediction for $t+1$. Similarly, the dual update is:

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta} \in \Theta} \left\{ q_{1:t}(\boldsymbol{\theta}) + \boldsymbol{\theta}^\top (\boldsymbol{m}_{1:t} + \widetilde{\boldsymbol{m}}_{t+1}) \right\}, \quad \text{with } \quad q_{1:t}(\boldsymbol{\theta}) = \frac{\sigma \|\boldsymbol{\theta}\|^2}{2} \sqrt{\sum_{\tau=1}^{t} \|\boldsymbol{m}_\tau - \widetilde{\boldsymbol{m}}_\tau\|_2^2} \tag{29}$$

where $\boldsymbol{m}_t = \nabla_{\boldsymbol{\theta}} \Psi_t^c(\boldsymbol{\theta}_t, \boldsymbol{y}_t)$. The detailed steps of the method are outlined in Algorithm 2, which follows the same template as Algorithm 1, sans the proxy function and the gradient definition (and its prediction) in the primal space. The regret of Algorithm 2 is summarized next.

Theorem 5.1. *Algorithm 2 attains regret:*

$$\mathcal{R}_T(G_\alpha) \leq \frac{4\sqrt{2} D_{\mathcal{Y}}}{T} \sqrt{\sum_{t=1}^{T} \|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|_2^2} + \frac{4\sqrt{2} D_\Theta}{T} \sqrt{\sum_{t=1}^{T} \|\boldsymbol{m}_t - \widetilde{\boldsymbol{m}}_t\|_2^2} + \frac{1}{T} \sum_{t=1}^{T} \left(\boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_T\right)^\top \boldsymbol{u}_t(\boldsymbol{y}^\star)$$

where $D_\Theta = \left(\frac{1}{u_{min}^{\alpha}} - \frac{1}{u_{max}^{\alpha}}\right)\sqrt{I}$, $\bar{\boldsymbol{\theta}}_T = (1/T) \sum_{t=1}^{T} \boldsymbol{\theta}_t$, and $D_{\mathcal{Y}}$ is the diameter of set $\mathcal{Y}$.

Discussion. The regret bound in the above Theorem verifies that the proposed OFTRL framework can deliver, also for this scenario, the desirable performance. We see that the first two regret terms shrink proportionally to the prediction errors and in any case do not exceed $O(\sqrt{T})$. On the other hand, the residual last term captures the perturbation of the dual variables from their respective horizon-long average value, modulated by the optimal utility vector and depends on the adversary strategy, cf. discussion of Theorem 4.3 and [81]. The execution of Algorithm 2 is lightweight as one can readily devise closed-form updates similar to those presented in Sec. 3, and, as such, suitable for the near-RT RIC. Finally, it is worth stressing that one can extend the above model by scalarizing the two criteria, i.e., weighting the two metrics so as to reflect the operational priorities w.r.t. fairness of performance for the users versus the energy cost of the vBS. This scalarization serves also the purpose of unifying the units of measure. We elaborate further on this aspect in Sec. 6.

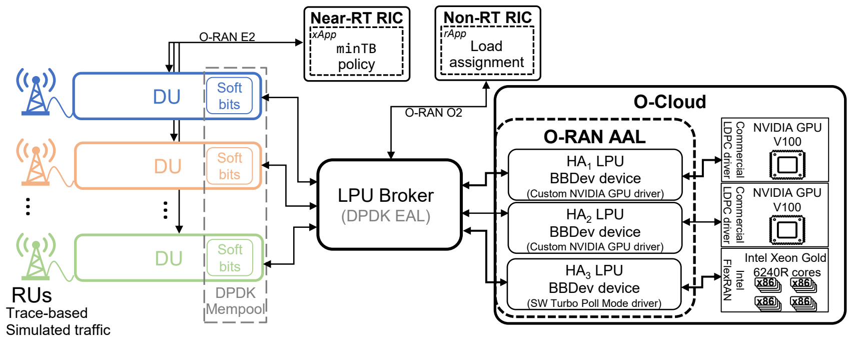

Near-RT RIC
xApp
minTB policy
Non-RT RIC
rApp
Load assignment
O-RAN E2
O-RAN O2
DU
Soft bits
RUs
Trace-based Simulated traffic
DPDK Mempool
LPU Broker
(DPDK EAL)
O-Cloud
O-RAN AAL
HA1 LPU BBDev device (Custom NVIDIA GPU driver)
HA2 LPU BBDev device (Custom NVIDIA GPU driver)
HA3 LPU BBDev device (SW Turbo Poll Mode driver)
Commercial LDPC driver
NVIDIA GPU V100
Intel FlexRAN
Intel Xeon Gold 6240R cores
x86

Fig. 5. Schematic of the experimental platform, including the RICs and interfaces of the use cases.

6 PERFORMANCE EVALUATION

We evaluate the proposed algorithms in a range of scenarios under realistic conditions. First, we use a simulator to assess the regret and performance - cost trade-off in these problems. The simulator uses traffic traces obtained from a real-world operational network and employs utility and cost functions that are built using measurements. Secondly, we implement the algorithms in an O-RAN-compliant experimental platform that follows the design principles in [78]. Thus, we measure the actual energy consumption and the processing latency of different baseband processors: two HAs and a pool of CPU cores. The platform uses two Nvidia GPU V100 as HAs and implements the O-RAN Acceleration Abstraction Layer (AAL) using Intel DPDK BBDev[10] according to specifications [3]. The AAL abstracts the O-Cloud computing resources as Logical Processing Units (LPUs). Note that the HAs consist of PCI boards which, although being faster in processing the workloads, they incur additional latency to transfer data from the software controller to the HA through a PCI bus, [57]. This latency is accounted for in our experimental setup, as it is part of the GPU processing time. For the CPU, we use an Intel Xeon Gold 6240R CPU with 32 cores, where 16 of them are assigned to signal processing tasks.

We generate the user traffic following the pattern from traces collected from a real BS using [36]. Based on this, we generate the TBs, modulate them according to 5G specifications, add noise based on the SNR of the traces, and finally inject them into the system. The platform processes the incoming signals using the open-access software library Intel FlexRAN [48]. We measure the energy consumption using the drivers of each PU, i.e., `RAPL` and `nvidia-smi` for the CPU and GPU, respectively. Fig. 5 presents a schematic of our experimental platform. Finally, we note that in O-RAN architecture [42], the non-RT and near-RT RICs operate closed-loops at, respectively, >1 second and 10-100 millisecond timescales. These timescales indicate how often the controller shall enforce a new policy [53]. To comply with such requirements, the application of the policy needs to be performed within a time window smaller than the timescale of the RICs. We confirm that all our algorithms require a negligible amount of time to execute (<10 ms), rendering them suitable to operate in the O-RAN RICs.

6.1 Load Assignment Control Policy

This section evaluates the vBSs' load assignment policy, which can be implemented as an rApp with a non-RT RIC at the SMO framework, and refers to a timescale of 1 second.

[10]https://doc.dpdk.org/guides/prog guide/bbdev.html

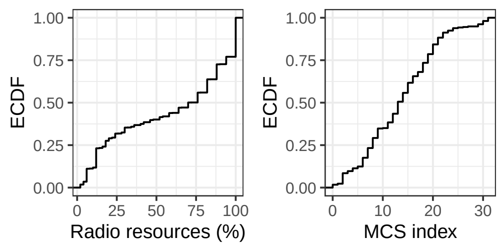


Fig. 6. Cell load dynamics collected from an operational RAN in Frankfurt, Germany, May 2023.

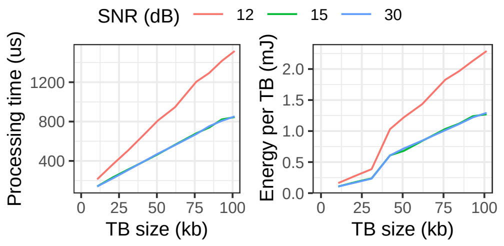


Fig. 7. Processing time (left) and energy consumption (right) when processing one TB with different sizes measured on an Intel Xeon CPU core.

6.1.1 Experimental Motivation. In Fig. 6 we delve into the traffic trace (see also Fig. 1), to observe the high variability of the allocated radio resources and network conditions (evidenced from MCS) in a single cell. This highlights the importance of RIC control policies to be adaptive, a need that becomes even more crucial in small and/or mobile cells. Secondly, Fig. 7 presents the processing time and energy cost when a CPU server processes one TB, for different TB sizes and SNRs. Comparing these results with those in Fig. 9, we find that CPU spends less energy per TB compared to a HA, especially for low SNRs, but this cost increases substantially with the TB size (amount of data). These findings highlight the potential benefits of an intelligent load assignment policy.

6.1.2 Simulation Study. We consider a simple model where the vBS utility[11] increases linearly with its load that is decoded at the assigned server, as long as the server is not overloaded, and it decreases rapidly when the server is assigned load that exceeds its capacity. In particular, the utility each vBS $i \in \mathcal{I}$ receives when sending $x_{ijt}\lambda_{it}$ load to a server $j \in \mathcal{J}$, is:

$$u_{ijt}(\boldsymbol{x}_t) = x_{ijt}\lambda_{it} \cdot \min\left\{1,\ 1 - \frac{1}{C_{jt}}\left(\sum_{k\in\mathcal{I}} \frac{x_{kjt}\lambda_{kt}}{n_{kt}}(\zeta_{kt}^j n_{kt} + o_{kt}^j) - C_{jt}\right)\right\}$$

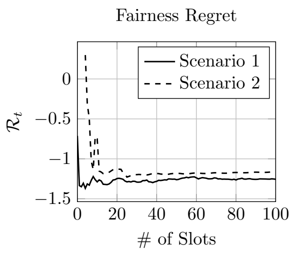
where λ_{kt} are the bytes sent by vBS $k \in \mathcal{J}$ during t and n_{kt} the average TB size of the flow (across all users). Parameters ζ_{kt}^j and o_{kt}^j model the slope and intercept for the processing time of server j, for the (average) SNR of vBS k during t; and we note that $\zeta_{kt}^j \approx 0$ for HA-based servers[12]. These parameters are obtained by fitting measurements as those in Fig. 7(left). Essentially the parenthesis term assess the portion of time that exceeds the server capacity, which we use to calculate how much vBS data are not decoded. Note that we use a more coarse-grained estimation for the number of expected TBs here than the respective expression in Sec. 6.2, due to the aggregation over longer time periods (1 sec instead of 100 msecs) and over multiple base stations.

For the cost function, we study the general case where the monetary energy cost can be different for each server, and we define the respective price vector $\boldsymbol{p}_t = (p_{jt} > 0, j \in \mathcal{J})$ (cost/J). Based on our experiments, we define a different (average) energy saving function for each server type:

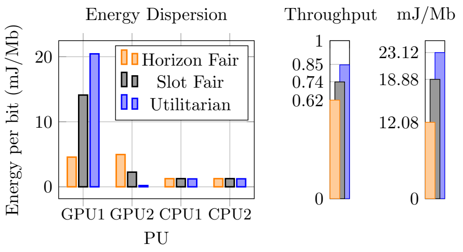
$$h_{jt}(\boldsymbol{x}_{jt}) = \varphi_h p_{jt} \sum_{i\in\mathcal{I}} \frac{(1 - x_{ijt})\lambda_{it}}{n_{it}} \cdot \left(\delta_{it}^j n_{it} + \gamma_{it}^j\right), \quad j \in \mathcal{J}. \tag{30}$$

Parameters δ_{it}^j and γ_{it}^j are the slope and intercept of the energy consumption profiles in Fig. 7(right) and Fig. 9(a-right), and φ_h is the normalization parameter and it is set by the operator to prioritize

[11] $u_{ijt}(\boldsymbol{x}_t)$ is concave on $\boldsymbol{x}_t$, see Appendix.
[12] The values of these parameters can be non-zero (but still very small) for certain TB value ranges. The algorithm and analysis are readily applicable to those cases, as well.



(a)



(b)

$(\alpha,\ \beta)$	mJ/Mb	Thr.
1, 1	41.92	0.851
1, 2	12.08	0.624
2, 2	40.00	0.776
0, 0	23.12	0.847

(c)

Fig. 8. **(a):** Fairness regret of Algorithm 1 for different number of slots T, averaged over 5 runs. **(b):** Energy dispersion amongst PUs, average throughput per vBS, and energy consumption per bit under the Horizon-Fair (Algorithm 1, $\alpha = 1, \beta = 2$), Slot-Fair ($\alpha = 1, \beta = 2$) and Utilitarian Algorithm ($\alpha = \beta = 0$). **(c):** PU energy consumption (mJ/Mbit) & Throughput (ratio of decoded TBs), for various α and β values.

throughput or energy. For HA servers, the energy depends on the number of TBs and their average SNR ($\delta_{it}^{j} \approx 0$); while for legacy CPU servers, it also depends on the TB size. Recall that we define h_{jt} as a cost reduction (energy savings) function, hence it is calculated w.r.t. the maximum possible cost for each server, i.e., when it serves all demand.

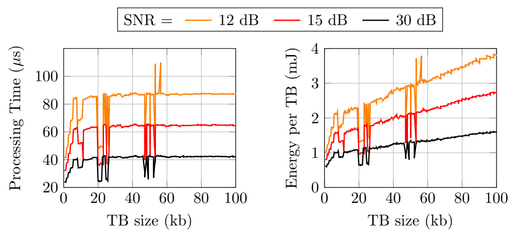
We consider the following two scenarios where we simulate a stationary and a non-stationary environment by setting $\lambda_{it}, n_{it}, p_{jt}, C_{jt}$, and average SNR s_{it}, $\forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T}$, as:

- *Scenario 1 (Synthetic, Stationary).* $I = 5$ vBSs and $J = 4$ servers, and the parameters are drawn randomly from uniform distributions: $\lambda_{it} \sim \mathcal{U}[4 \cdot 10^6, 6 \cdot 10^6)$, $n_{it} \sim \mathcal{U}[4 \cdot 10^4, 6 \cdot 10^4)$, $s_{it} \sim \mathcal{U}[10, 30)$, $p_{jt} \sim \mathcal{U}[10, 15)$, and $C_{jt} \sim \mathcal{U}[0, 10)$. We set $\alpha = \beta = \varphi_h = 1$, unless stated otherwise.
- *Scenario 2 (Synthetic, Non-stationary).* $\boldsymbol{C}_t$ follows a periodic pattern, while $\boldsymbol{\lambda}_t, \boldsymbol{n}_t, \boldsymbol{s}_t$, and $\boldsymbol{p}_t$ have vanishing perturbations. We draw the mean values for $\boldsymbol{C}_t$, $\boldsymbol{\lambda}_t, \boldsymbol{n}_t, \boldsymbol{s}_t$, and $\boldsymbol{p}_t$ from $\mathcal{U}[0, 10)$, $\mathcal{U}[4 \cdot 10^6, 6 \cdot 10^6)$, $\mathcal{U}[4 \cdot 10^4, 6 \cdot 10^4)$, $\mathcal{U}[10, 30)$, and $\mathcal{U}[10, 15)$, respectively; and we perturb $\boldsymbol{C}_t$ with a sine wave of period $\sqrt{T}$, $\boldsymbol{\lambda}_t, \boldsymbol{n}_t, \boldsymbol{s}_t$ with vanishing Gaussian noise scaled with t^{-1}, and $\boldsymbol{p}_t$ with vanishing Gaussian noise scaled with $0.1t^{-1}$. We set $\alpha = \beta = \varphi_h = 1$. $\boldsymbol{C}_t$ has sublinear partitioned severity and the other parameters have sublinear budgeted severity.

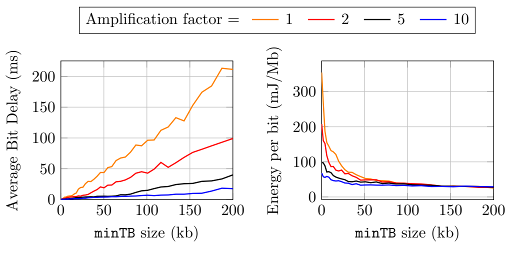
We first estimate the intercept and slope parameters $\zeta_{it}^{j}, o_{it}^{j}, \delta_{it}^{j}$, and γ_{it}^{j} using the measurements obtained from the testbed and linear regression for each SNR value; and then use Algorithm 1. The horizon-fair regret is shown in Fig. 8(a), aggregated over 5 independent runs. Aligned with the theoretical analysis, the algorithm achieves sublinear (in fact, negative) fairness regret in both the stationary and non-stationary scenarios. Indeed, we observe the convergence in these experiments is particularly fast, as it requires only a few tens of slots to reach the performance of the benchmark.

6.1.3 Experimental Evaluation. Next, we evaluate the algorithm on an O-RAN compliant platform [78]. We consider a setting with 5 identical vBSs with 100 users each. As explained above, the traffic generation and SNR patterns are based on traffic traces collected from real BSs, we scale the energy saving function by setting $\varphi_h = \max\{\lambda_{it}\}/\max\{p_{jt}\}$ so that both the utility ($u_{it}(\boldsymbol{x}_t)$) and energy saving ($h_{jt}(x_{jt})$) functions are scaled between 0 and $\max\{\lambda_{it}\}$. We measure the PUs energy and normalized throughput (ratio of successfully decoded TBs) from the experimental platform at TTI granularity and aggregate the measurements to produce decisions at the non-RT timescale. In order to emulate heterogeneous HAs, we half the speed of the second GPU (using the Nvidia drivers) and artificially double its energy cost in our measurements. We also consider two identical CPUs.

For comparison, we implement two new algorithms, namely a slot-fair algorithm, in line with suggestions in [50, 82, 83]; and an algorithm that maximizes the aggregate system utility (utilitarian),



(a)



(b)

Fig. 9. **(a):** Experimental measurement of computing time and energy consumption per TB, for different TB sizes. **(b):** Energy consumption per bit and average bit delay for a GPU HA as a function of the TB size threshold (`minTB`) and for different amplification factors applied to the traffic traces.

without catering for any type of fairness. Namely, the objective of the slot-fair algorithm is to maximize the fairness in each slot, whereas the objective of the utilitarian algorithm is to maximize the sum of HAs' energy savings and vBSs' utilities. To simplify the comparison, we use the non-optimistic versions of our algorithm.

The Table in Fig. 8(c) summarizes the trade-off between the average throughput per vBS and the PUs' energy consumption when we impose the fairness criteria. As expected, the utilitarian algorithm (i.e., $\alpha = \beta = 0$) outperforms Algorithm 1 in regards to total throughput and energy. However, Fig. 8(b) clearly shows that the utilitarian solution directs most demand to GPU1 and the CPUs, and does not employ GPU2 which is intentionally designed to be slower and more energy-consuming in this scenario. That is, the utilitarian solution allows the maximization of energy savings and throughput by not using the worse GPU, since there are no fairness requirements. In contrast, the fair algorithms direct a significant portion of the vBSs demand to GPU2, increasing the energy consumption. The horizon fair algorithm is more fair than the slot fair algorithm with respect to energy dispersion amongst HAs, ending up almost equalizing the energy consumption of both GPUs. Fig. 8(c) also indicates that modifying α and β parameters has an unintentional effect on the prioritization of different objectives. Due to the exponential nature of $\alpha-$fairness function, increasing α prioritizes the utilities more, and results in more throughput; while increasing β prioritizes the energy savings more, which results in reduced energy consumption. The decision of φ_h should be made attentively to prevent any side effect when modifying the fairness parameters.

6.2 `minTB` Control Policy

Next, we evaluate the near-real-time compute control policy `minTB` that can be applied to each vBS independently. As in the previous section, we provide experimental motivation for the problem, run simulations with traces, and implement the solution at an O-RAN testbed.

6.2.1 Experimental Motivation. Fig. 9(a) presents the processing time and energy consumption of a GPU server (a common HA) for different TBs and SNRs. The processing time is practically independent of the TB size, an advantage stemming from the GPU's parallelization capability. Similarly, the energy consumption increases only slightly with the TB size. For example, with 15 dB SNR, the energy cost for 20 and 100 kb TBs (5× increase) is 1.7 and 2.8 mJ respectively (0.5× increase)[13]. Nevertheless, users often transmit TBs of small length, see [38] and our traces, thus inducing unnecessary energy costs. The `minTB` policy can tackle this issue. Indeed, in the experiments presented in Fig. 9(b), we see how the `minTB` value affects the vBS energy and the

[13]This small increase arises for very large load increments that require engaging additional processing elements of the GPU.

delay for users with different loads. For example, with a minimum TB size of 25 kb, the energy consumption drops up to 4× and the delay increases up to 20 msec, compared to when not using any threshold, which is currently the default implementation of software-defined base stations.

The `minTB` policy can be implemented as an xApp in the Near-RT RIC, which operates in slots of ∼100 ms. The users send data at each TTI (every 1 ms) and can provide feedback about their buffer status at such fine granularity. The algorithm selects the `minTB` value for each user and communicates (via the O1 interface) this rule to vBS, which is enforced by the radio scheduler at every TTI. The policy is updated every slot (100 TTIs), based on the users' feedback and experienced delay (calculated by the RIC), and the reported vBS energy consumption during the previous slots.

6.2.2 Simulation Study. The algorithms require a model for the utility and cost functions which we build using experimental results. Let us denote with b_{it} the number of *data generation events* of user i during each slot $t \in \mathcal{T}$, and with ρ_{it} the average number of bits generated at each such event, and define $\boldsymbol{b}_t = (b_{it}, i \in \mathcal{I})$, $\boldsymbol{\rho}_t = (\rho_{it}, i \in \mathcal{I})$. We assume that these values follow a Poisson distribution during each slot, but can change arbitrarily across different slots. Hence, we employ the following approximation for the HA energy cost:

$$c_t(\boldsymbol{y}) = \varphi \sum_{i \in \mathcal{I}} \beta(s_{it})\pi_{it} = \varphi \sum_{i \in \mathcal{I}} \beta(s_{it}) b_{it} \frac{\rho_{it}}{y_i}\left(1 - e^{-\frac{y_i}{\rho_{it}}}\right), \tag{31}$$

where π_{it} is the expected number of TBs user i will generate in slot t, $\beta(\cdot)$ is the mapping from SNR to a cost coefficient (as SNR affects the energy cost); s_{it} is the average SNR of user i in t (calculated at the end of the slot); and φ a normalization parameter that can prioritize cost over fairness, if necessary. For the utility function, we use the percentage of time the user's buffer is empty [16]. This metric acts as a proxy for the delay. As the network is not in saturation most of the time, we assume that each user empties its buffer as soon as its data exceeds the TB threshold. Based on that, we derive the following approximation:

$$u_{it}(\boldsymbol{y}) = \mathbf{Pr}(B_{it} = 0) = \frac{\rho_{it}}{y_i}\left(1 - e^{-\frac{y_i}{\rho_{it}}}\right), \tag{32}$$

where B_{it} is the number of bits in the buffer of user i in time slot t, and $\mathbf{Pr}(B_{it} = 0)$ the probability of empty buffer. Users experience more delay and the HA energy incurred by the user decreases as the value of the TB threshold increases. The approximations (31) and (32) captures this dependency and both the cost and utility functions are decreasing functions. We validate these functions using real data gathered from the O-RAN platform (see Appendix).

We consider the following three scenarios for the simulations:

- *Scenario 1 (Synthetic, Stationary).* The vBS serves $I = 10$ users, and the parameters are uniformly random as $b_{it} \sim \mathcal{U}(10, 40)$, $\rho_{it} \sim \mathcal{U}(5 \cdot 10^4, 10^5)$, $s_{it} \sim \mathcal{U}(20, 30)$, $\forall i \in \mathcal{I}, t \in \mathcal{T}$.
- *Scenario 2 (Traced-driven).* We use the above traces from a vBS obtained with [36], and generate the values for b_{it}, ρ_{it}, and s_{it} for 5 users whose data parameters (i.e., ρ_{it}) are scaled by 1, 2, 4, 6 and 8.
- *Scenario 3 (Synthetic, non-stationary).* We consider $I = 5$ users with data generation and SNR values that follow an adversarial *ping-pong* pattern which, further, is different for each user:

$$b_{it} = \begin{cases} 10 & \text{if } t < 2^{i-1} \pmod{2^i} \\ 40 & \text{otherwise} \end{cases}, \qquad s_{it} = \begin{cases} 20 & \text{if } t < 2^{4-i} \pmod{2^{5-i}} \\ 30 & \text{otherwise} \end{cases}.$$

We restrict the adversary to have sublinear budgeted severity and set $\boldsymbol{\rho}_t$ as $\rho_{it} = \bar{\rho}_i + 10^4 n_{it}/t$ where $\bar{\rho}_i \sim \mathcal{U}(5 \cdot 10^4, 10^5)$ and $n_{it} \sim \mathcal{N}(0, 1)$.

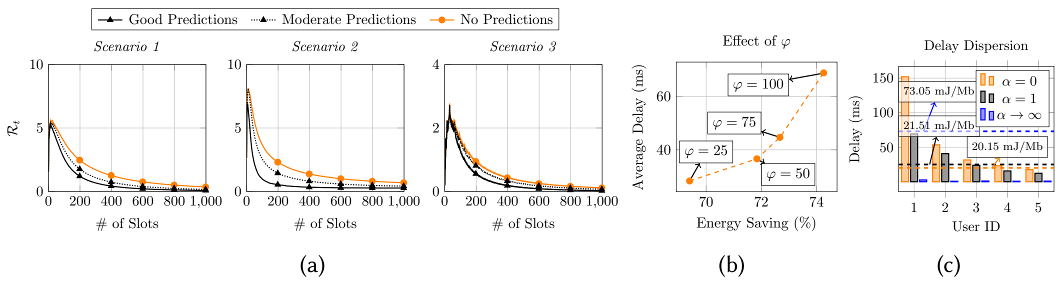


Fig. 10. **(a):** Regret $\mathcal{R}_T(G_\alpha)$ of Algorithm 2 for different number of slots T, averaged over 10 runs. **(b):** Energy savings and user delay (performance) for different values of φ, and $\alpha = 1$. **(c) :** Delay of each user with respect to α and HA's energy consumption per bit (dashed lines), for $\varphi = 50$.

The fairness parameter is set to $\alpha = 1$, unless stated otherwise. Finally, we consider two types of predictions: good and moderate predictions. We obtain the prediction of the gradients in step 8 of Algorithm 1 by first calculating the actual gradient in the next slot and then adding a Gaussian noise, scaled with the gradient and accuracy coefficient. For the good predictions, the accuracy coefficient is 0.001, whereas for moderate predictions, we set the coefficient to 0.3. For instance, the good prediction of $\tilde{\boldsymbol{g}}_{t+1}$ is calculated as $\tilde{\boldsymbol{g}}_{t+1} = \boldsymbol{g}_{t+1} + 0.001\boldsymbol{n}_{t+1} \circ \boldsymbol{g}_{t+1}$ where $\boldsymbol{n}_{t+1}$ is Gaussian noise.

Fig. 10(a) plots the regret $\mathcal{R}_T(G_\alpha)$ for the above scenarios and prediction models. In Scenario 1 (left), Algorithm 2 converges independently of the quality of predictions. In Scenario 2 (center), the algorithm with good predictions achieves lower regret and converges faster, as expected. However, due to high variations in the utility values of these traces (unrestricted adversary), the residual (last) term in Theorem 5.1 is not eliminated. Finally, Fig. 10(a-right) shows the results for the restricted adversarial scenario where all algorithms achieve zero regret for $T > 1000$.

6.2.3 Experimental Evaluation. Next, we implement and evaluate Algorithm 2 in a testbed. The `minTB` policy $\boldsymbol{y}_t$ is derived using the cost and utility models (31)-(32), and then implemented in the platform where we measure the actual energy consumption and (average) delay for each user.

First, we evaluate the effect of φ which balances the importance of energy cost and user utility. In Fig. 10(b), we compare the energy consumption of the `minTB` policies with the default policy where no threshold is applied to TBs, i.e., $\boldsymbol{y} = \boldsymbol{0}$ (current default in such vBS). We calculate the energy saving of the `minTB` policy with respect to the default policy and plot the average measured delays. As φ increases, the energy savings improve alongside an increase in average delay. We see, for instance, that we can save a remarkable amount of 67% energy compared to the case no TB threshold is used, at the expense of ~15msec additional average delay for the users; and we can save up to 72% energy without incurring more than ~38msec delay, by tuning the control parameter φ accordingly. Next, we showcase how this delay is dispersed across the users. Fig. 10(c) plots the average (over time) delay per user in Scenario 2 with 5 users for $\alpha = 0$ (aggregate delay minimization), $\alpha = 1$ (proportional fairness), and $\alpha \longrightarrow \infty$ (max-min fairness). Since the data of each user might induce different energy costs due to their SNR and/or volume, the RIC will naturally apply a different `minTB` policy per user, hence inducing a different delay for each of them. The value of α affects these decisions directly. Indeed, we see that the delay dispersion is more fair when we set $\alpha = 1$ and $\alpha \longrightarrow \infty$; while the latter creates $>3\times$ more energy consumption.

7 CONCLUSIONS

O-RAN, and similar virtualized RAN architectures, promise unprecedented performance and versatility for next generation of mobile networks, yet their energy costs are likely to constitute a prohibitive deployment factor. Motivated by this, we propose a radio-control policy and a compute-control (assignment) policy which cater for the energy consumption of vBSs and their O-Cloud processing units. The policies balance the user-perceived performance (throughput and transmission delay) with the network's energy costs, and importantly, disperse them *fairly* across the users and the servers (respectively) throughout the entire operation of the system. The decision engine of the policies utilizes online learning algorithms (optimistic FTRL) that are tailored for the problem at hand, and as such is robust to a wide range of (unpredictable) parameter perturbations. We prove and demonstrate the optimality of these algorithms using a range of scenarios, both with simulations and testbed experiments, and measure energy savings (per vBS) up to 72% when the users can tolerate $\sim$ 38msec additional delay, on average.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers and Igor Kadota (shepherd) for their valuable feedback that helped us improve this work. This work has been supported by the European Commission through Grant No. SNS-JU-101097083 (BeGREEN), 101139270 (ORIGAMI), and 101017109 (DAEMON) and CERCA Programme.

REFERENCES

[1] Shipra Agrawal and Nikhil Devanur. 2014. Bandits with Concave Rewards and Convex Knapsacks. In *Proceedings of ACM EC*. 989–1006.

[2] Juan J Alcaraz, Jose A Ayala-Romero, Javier Vales-Alonso, and Fernando Losilla-López. 2020. Online Reinforcement Learning for Adaptive Interference Coordination. *Transactions on Emerging Telecommunications Technologies* 31, 10 (2020), e4087.

[3] O-RAN ALLIANCE. 2021. O-RAN Acceleration Abstraction Layer General Aspects and Principles. O-RAN.WG6.AAL-GAnP-v01.00.

[4] Ismail Alqerm and Basem Shihada. 2018. Sophisticated Online Learning Scheme for Green Resource Allocation in 5G Heterogeneous Cloud Radio Access Networks. *IEEE Transactions on Mobile Computing* 17, 10 (2018), 2423–2437.

[5] Eitan Altman, Konstantin Avrachenkov, and Andrey Garnaev. 2008. Generalized α-fair Resource Allocation in Wireless Networks. In *Proceedings of IEEE CDC*. 2414–2419.

[6] Eitan Altman, Konstantin Avrachenkov, and Sreenath Ramanath. 2012. Multiscale Fairness and its Application to Resource Allocation in Wireless Networks. *Computer Communications* 35, 7 (2012), 820–828.

[7] Analysys Mason. 2023. Key TCO Considerations for Economically Viable Open RAN . Strategy report.

[8] Analysys Mason. 2023. Open RAN: Translating the Hype into Revenue. Webinar.

[9] Daren Anderson, George Iosifidis, and Douglas Leith. 2023. Lazy Lagrangians for Optimistic Learning with Budget Constraints. *IEEE/ACM Transactions on Networking* 31, 5 (2023), 1935–1949.

[10] GSMA Association. 2020. 5G Energy Efficiencies: Green is the New Black. *White Paper* (2020).

[11] Gunther Auer, Vito Giannini, Claude Desset, Istvan Godor, Per Skillermark, Magnus Olsson, Muhammad Ali Imran, Dario Sabella, Manuel J Gonzalez, Oliver Blume, et al. 2011. How Much Energy is Needed to Run a Wireless Network? *IEEE wireless communications* 18, 5 (2011), 40–49.

[12] Peter Auer, Nicolo Cesa-Bianchi, and Claudio Gentile. 2002. Adaptive and Self-confident Online Learning Algorithms. *J. Comput. System Sci.* 64 (2002), 48–75.

[13] Jose A Ayala-Romero, Juan J Alcaraz, Andrea Zanella, and Michele Zorzi. 2019. Online Learning for Energy Saving and Interference Coordination in Hetnets. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1374–1388.

[14] Jose A Ayala-Romero, Andres Garcia-Saavedra, Xavier Costa-Perez, and George Iosifidis. 2021. EdgeBOL: Automating Energy-Savings for Mobile Edge AI. In *Proceedings of ACM CoNEXT*. 397–410.

[15] Jose A Ayala-Romero, Andres Garcia-Saavedra, Xavier Costa-Perez, and George Iosifidis. 2021. Orchestrating Energy-Efficient vRANs: Bayesian Learning and Experimental Results. *IEEE Transactions on Mobile Computing* 22, 5 (2021), 2910–2924.

[16] Jose A Ayala-Romero, Andres Garcia-Saavedra, Marco Gramaglia, Xavier Costa-Perez, Albert Banchs, and Juan J Alcaraz. 2019. vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs. In *Proceedings of MobiCom*. 1–16.

[17] Jose A Ayala-Romero, Andres Garcia-Saavedra, Marco Gramaglia, Xavier Costa-Pérez, Albert Banchs, and Juan J Alcaraz. 2022. vrAIn: Deep Learning Based Orchestration for Computing and Radio Resources in vRANs. *IEEE Transactions on Mobile Computing* 21, 7 (2022), 2652–2670.

[18] Jose A Ayala-Romero, Ihtisham Khalid, Andres Garcia-Saavedra, Xavier Costa-Perez, and George Iosifidis. 2021. Experimental Evaluation of Power Consumption in Virtualized Base Stations. In *Proceedings of IEEE ICC*. 1–6.

[19] Jackie Baek and Vivek Farias. 2021. Fair Exploration via Axiomatic Bargaining. *Proceedings of NeurIPS*, 22034–22045.

[20] Amir Beck. 2017. First-Order Methods in Optimization. *MOS-SIAM Series on Optimization* (2017).

[21] Dario Bega, Albert Banchs, Marco Gramaglia, Xavier Costa-Pérez, and Peter Rost. 2018. CARES: Computation-aware Scheduling in Virtualized Radio Access Networks. *IEEE Transactions on Wireless Communications* 17, 12 (2018), 7993–8006.

[22] Dario Bega, Marco Gramaglia, Marco Fiore, Albert Banchs, and Xavier Costa-Perez. 2019. DeepCog: Optimizing Resource Provisioning in Network Slicing with AI-based Capacity Forecasting. *IEEE Journal on Selected Areas in Communications* 38, 2 (2019), 361–376.

[23] Gerdus Benade, Aleksandr M Kazachkov, Ariel D Procaccia, and Christos-Alexandros Psomas. 2018. How to Make Envy Vanish Over Time. In *Proceedings of ACM EC*. 593–610.

[24] Dimitri P Bertsekas. 2016. *Nonlinear Programming, 3rd Edition.* Athena Scientific.

[25] Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. 2011. The price of fairness. *Operations research* 59, 1 (2011), 17–31.

[26] Yufei Blankenship, Dennis Hui, and Mattias Andersson. 2021. *Channel Coding in NR.* Springer International Publishing, Cham, 303–332. https://doi.org/10.1007/978-3-030-58197-8_10

[27] Thomas Bonald and James W Roberts. 2015. Multi-Resource Fairness: Objectives, Algorithms and Performance. In *Proceedings of ACM Sigmetrics*. 31–42.

[28] Leonardi Bonati, Salvatore D'Oro, Michele Polese, Stefano Basagni, and Tommaso Melodia. 2021. Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks. *IEEE Commun. Mag.* 59, 10 (2021), 21–27.

[29] Niv Buchbinder and Joseph Naor. 2013. Fair Online Load Balancing. *Journal of Scheduling* 16 (2013), 117–127.

[30] Semih Cayci, Swati Gupta, and Atilla Eryilmaz. 2020. Group-fair Online Allocation in Continuous Time. *Proceedings of NeurIPS* 33, 13750–13761.

[31] Sina Darabi, Negin Mahani, Hazhir Bakhishi, Ehsan Yousefzadeh-Asl-Miandoab, Mohammad Sadrosadati, and Hamid Sarbazi-Azad. 2022. NURA: A Framework for Supporting Non-Uniform Resource Accesses in GPUs. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 1 (2022), 16:1–16:27.

[32] Dell. 2022. Dell Open RAN Accelerator Card. In *Solution Brief.*

[33] Salvatore D'Oro, Michele Polese, Leonardo Bonati, Hai Cheng, and Tommaso Melodia. 2022. dApps: Distributed Applications for Real-Time Inference and Control in O-RAN. *IEEE Commun. Mag.* 60, 11 (2022), 52–58.

[34] Ericsson. 2023. Ericsson Unwraps New Energy Efficiency Solutions Designed for Open RAN Architecture. https://www.ericsson.com/en/news/2023/4/energy-efficient-ericsson-rapps-for-open-ran-architecture.

[35] Gabriel Falcao, Leonel Sousa, and Vitor Silva. 2010. Massively LDPC Decoding on Multicore Architectures. *IEEE Transactions on Parallel and Distributed Systems* 22, 2 (2010), 309–322.

[36] Robert Falkenberg and Christian Wietfeld. 2019. FALCON: An Accurate Real-time Monitor for Client-based Mobile Network Data Analytics. In *Proceedings of IEEE GLOBECOM.* IEEE, 1–7.

[37] Francesca Fossati, Stefano Moretti, Patrice Perny, and Stefano Secci. 2020. Multi-resource Allocation for Network Slicing. *IEEE/ACM Transactions on Networking* 28, 3 (2020), 1311–1324.

[38] Xenofon Foukas and Bozidar Radunovic. 2021. Concordia: Teaching the 5G vRAN to Share Compute. In *Proceedings of ACM SIGCOMM.* 580–596.

[39] Apostolos Galanopoulos, Jose A Ayala-Romero, George Iosifidis, and Douglas Leith. 2020. Bayesian Online Learning for MEC Object Recognition Systems. In *Proceedings of IEEE GLOBECOM.* IEEE.

[40] Gines Garcia-Aviles, Andres Garcia-Saavedra, Marco Gramaglia, Xavier Costa-Perez, Pablo Serrano, and Albert Banchs. 2021. Nuberu: Reliable RAN Virtualization in Shared Platforms. In *Proceedings of ACM MobiCom.* 749–761.

[41] Andres Garcia-Saavedra, Xavier Costa-Perez, Douglas Leith, and George Iosifidis. 2018. Fluidran: Optimized VRAN/MEC Orchestration. In *Proceedings of IEEE INFOCOM.* 2366–2374.

[42] Andres Garcia-Saavedra and Xavier Costa-Pérez. 2021. O-RAN: Disrupting the Virtualized RAN Ecosystem. *IEEE Communications Standards Magazine* 5, 4 (2021), 96–103.

[43] Leonidas Georgiadis, Michael J Neely, and Leandros Tassiulas. 2006. Resource Allocation and Cross-Layer Control in Wireless Networks. *Foundations and Trends in Networking* 1, 1 (2006).

[44] Swati Gupta and Vijay Kamble. 2021. Individual Fairness in Hindsight. *The Journal of Machine Learning Research* 22, 1 (2021), 6386–6420.

[45] Hassan Halabian. 2019. Distributed Resource Allocation Optimization in 5G Virtualized networks. *IEEE Journal on Selected Areas in Communications* 37, 3 (2019), 627–642.

[46] Elad Hazan. 2016. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization* 2, 3-4 (2016), 157–325.

[47] Rob J Hyndman and George Athanasopoulos. 2018. *Forecasting: Principles and Practice.* OTexts.

[48] Intel. 2019. FlexRAN LTE and 5G NR FEC Software Development Kit Modules. (2019).
[49] Intel. 2021. Virtual RAN (vRAN) with Hardware Acceleration. In *White Paper*.
[50] Devansh Jalota and Yinyu Ye. 2022. Online Learning in Fisher Markets with Unknown Agent Preferences. *arXiv preprint arXiv:2205.00825* (2022).
[51] Michail Kalntis and George Iosifidis. 2022. Energy-Aware Scheduling of Virtualized Base Stations in O-RAN with Online Learning. In *Proceedings of IEEE GLOBECOM*. 6048–6054.
[52] Frank Kelly, Aman K Maulloo, and David Kim Hong Tan. 1998. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness, and Stability. *J. Oper. Res. Soc.* 49, 3 (1998), 237–252.
[53] Woo-Hyun Ko, Ushasi Ghosh, Ujwal Dinesha, Raini Wu, Srinivas Shakkottai, and Dinesh Bharadia. 2023. Demo: EdgeRIC: Delivering Realtime RAN Intelligence. In *Proceedings of ACM SIGCOMM*. 1162–1164.
[54] Baolin Li, Tirthak Patel, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. 2022. MISO: Exploiting Multi-instance GPU Capability on Multi-tenant GPU Clusters. *Proceedings of ACM SoCC* (2022), 173–189.
[55] Luofeng Liao, Yuan Gao, and Christian Kroer. 2022. Nonstationary Dual Averaging and Online Fair Allocation. *Proc. of NeurIPS*, 37159–37172.
[56] China Mobile Limited. 2021. 2021 Sustainability Report. *White Paper* (2021).
[57] Dimitrios Mbakoyiannis, Othon Tomoutzoglou, and George Kornaros. 2018. Energy-performance considerations for data offloading to FPGA-based accelerators over PCIe. *ACM Transactions on Architecture and Code Optimization (TACO)* 15, 1 (2018), 1–24.
[58] H Brendan McMahan. 2017. A Survey of Algorithms and Analysis for Adaptive Online Learning. *The Journal of Machine Learning Research* 18, 1 (2017), 3117–3166.
[59] Fidan Mehmeti and Wolfgang Kellerer. 2022. Max-min Fair Resource Allocation in SD-RAN. In *Proceedings of ACM Q2SWinet*. 27–35.
[60] Fidan Mehmeti and Thomas La Porta. 2022. Reducing the Cost of Consistency: Performance Improvements in Next Generation Cellular Networks With Optimal Resource Reallocation. *IEEE Transactions on Mobile Computing* 21, 7 (2022), 2546–2565.
[61] Naram Mhaisen, George Iosifidis, and Douglas Leith. 2022. Online Caching with Optimistic Learning. In *Proceedings of IFIP Networking*.
[62] Naram Mhaisen, Abhishek Sinha, Georgios Paschos, and George Iosifidis. 2022. Optimistic No-regret Algorithms for Discrete Caching. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3 (2022), 1–28.
[63] Jeonghoon Mo and Jean Walrand. 2000. Fair End-to-end Window-based Congestion Control. *IEEE/ACM Transactions on Networking* 8, 5 (2000), 556–567.
[64] Naresh Modina, Mandar Datar, Rachid El-Azouzi, and Francesco de Pellegrini. 2022. Multi Resource Allocation for Network Slices with Multi-Level fairness. In *ICC 2022-IEEE International Conference on Communications*. IEEE, 4872–4877.
[65] Sharayu Moharir, Sujay Sanghavi, and Sanjay Shakkottai. 2015. Online Load Balancing Under Graph Constraints. *IEEE/ACM Transactions on Networking* 24, 3 (2015), 1690–1703.
[66] Mehryar Mohri and Scott Yang. 2016. Accelerating Online Convex Optimization via Adaptive Prediction. In *Proceedings of AISTATS*. 848–856.
[67] Sourav Mondal and Marco Ruffini. 2023. Fairness Guaranteed and Auction-Based x-Haul and Cloud Resource Allocation in Multi-Tenant O-RANs. *IEEE Transactions on Communications* 71, 6 (2023), 3452–3468.
[68] Dritan Nace and Michal Pioro. 2008. Max-min Fairness and its Applications to Routing and Load-balancing in Communication Networks: a Tutorial. *IEEE Communications Surveys & Tutorials* 10, 4 (2008), 5–17.
[69] Michael J Neely. 2010. Stochastic Network Optimization with Application to Communication and Queueing Systems. *Synthesis Lectures on Communication Networks* (2010).
[70] Michael J Neely. 2013. Delay-Based Network Utility Maximization. *IEEE/ACM Transactions on Networking* 21, 1 (2013), 41–54.
[71] Duong Tung Nguyen, Long Bao Le, and Vijay K Bhargava. 2019. A Market-based Framework for Multi-resource Allocation in Fog Computing. *IEEE/ACM Transactions on Networking* 27, 3 (2019), 1151–1164.
[72] Francesco Orabona. 2019. A Modern Introduction to Online Learning. *arXiv preprint arXiv:1912.13213* (2019).
[73] Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. 2023. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Commun. Surv. Tutorials* 25, 2 (2023), 1376–1411.
[74] Darijo Raca, Ahmed H Zahran, Cormac J Sreenan, Rakesh K Sinha, Emir Halepovic, Rittwik Jana, and Vijay Gopalakrishnan. 2020. On Leveraging Machine and Deep Learning for Throughput Prediction in Cellular Networks: Design, Performance, and Challenges. *IEEE Communications Magazine* 58, 3 (2020), 11–17.
[75] Bozidar Radunovic and Jean-Yves Le-Boudec. 2007. A Unified Framework for Max-Min and Min-Max Fairness With Applications. *IEEE/ACM Transactions on Networking* 15, 5 (2007), 1073–1083.

[76] Alexander Rakhlin and Karthik Sridharan. 2013. Optimization, Learning, and Games with Predictable Sequences. In *Proceedings of NIPS*. 848–856.
[77] Peter Rost, Andreas Maeder, Matthew C Valenti, and Salvatore Talarico. 2015. Computationally-aware Sum-rate Optimal Scheduling for Centralized Radio Access Networks. In *Proc. of IEEE GLOBECOM*. 1–6.
[78] J Xavier Salvat, Jose A Ayala-Romero, Lanfranco Zanzi, Andres Garcia-Saavedra, and Xavier Costa-Perez. 2023. Open Radio Access Networks (O-RAN) Experimentation Platform: Design and Datasets. *IEEE Communications Magazine* (2023).
[79] Shai Shalev-Shwartz. 2012. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning* 4, 2 (2012), 107–194.
[80] Shai Shalev-Shwartz and Yoram Singer. 2007. A Primal-dual Perspective of Online Learning Algorithms. *Machine Learning* 69, 2-3 (2007), 115–142.
[81] Tareq Si Salem, Georgios Iosifidis, and Giovanni Neglia. 2022. Enabling Long-term Fairness in Dynamic Resource Allocation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3 (2022), 1–36.
[82] Sean R Sinclair, Siddhartha Banerjee, and Christina Lee Yu. 2022. Sequential Fair Allocation: Achieving the Optimal Envy-Efficiency Tradeoff Curve. *Operations Research* 71, 5 (2022).
[83] Mohammad Sadegh Talebi and Alexandre Proutiere. 2018. Learning Proportionally Fair Allocations with Low Regret. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 2 (2018), 1–31.
[84] Telefonica. 2022. Open RAN Technical Priorities Release 2. https://www.telefonica.com/en/communication-room/reports/open-ran-technical-priorities-release-2/.
[85] Sharda Tripathi, Corrado Puligheddu, Somreeta Pramanik, Andres Garcia-Saavedra, and Carla Fabiana Chiasserini. 2023. Fair and Scalable Orchestration of Network and Compute Resources for Virtual Edge Services. *IEEE Transactions on Mobile Computing* early access (2023), 1–17.
[86] Ke Wang, XiaoYi Yu, WenLiang Lin, ZhongLiang Deng, and Xin Liu. 2021. Computing-aware Scheduling in Mobile Edge Computing System. *Wireless Networks* 27 (2021), 4229–4245.
[87] Wei Wang, Baochun Li, and Ben Liang. 2014. Dominant Resource Fairness in Cloud Computing Systems with Heterogeneous Servers. In *Proceedings of IEEE INFOCOM*. 583–591.
[88] Wentao Weng, Xingyu Zhou, and Rayadurgam Srikant. 2020. Optimal Load Balancing with Locality Constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3 (2020), 45:1–45:37.
[89] Jie Xu, Lixing Chen, and Shaolei Ren. 2017. Online Learning for Offloading and Autoscaling in Energy Harvesting Mobile Edge Computing. *IEEE Transactions on Cognitive Communications and Networking* 3, 3 (2017), 361–373.
[90] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. 2019. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Commun. Surv. Tutor.* 21, 3 (2019), 2224–2287.
[91] Nan Zhao, Ying-Chang Liang, Dusit Niyato, Yiyang Pei, Minghu Wu, and Yunhao Jiang. 2019. Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Cellular Networks. *IEEE Transactions on Wireless Communications* 18, 11 (2019), 5141–5152.
[92] Martin Zinkevich. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of ICML*. 928–936.

8 APPENDIX

This section provides the remaining proofs for the results presented in the previous sections, as well as some additional evaluation results for the interested reader. Please note that we abuse slightly the notation by redefining and reusing some symbols, in order to keep the presentation streamlined.

8.1 Proof of Lemma 4.1

This lemma applies to the dual update for $\boldsymbol{\theta}_{t+1}$ in (12) that uses the regularizers (14). Applying [66, Theorem 2], we can write:

$$\mathcal{R}_T^{\theta} - q_{1:T-1}(\boldsymbol{\theta}^\star) \leq \sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_{(t-1),*}^2 = \sum_{t=1}^{T} \frac{\|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}{\sigma_{1:t-1}} = \sum_{t=1}^{T} \frac{(1/\sigma)\|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}{\sqrt{\sum_{\tau=1}^{t-1} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}}. \tag{33}$$

Similarly, from the proof of the same Theorem, we extract the inequality:

$$\mathcal{R}_T^{\theta} - q_{1:T-1}(\boldsymbol{\theta}^\star) \leq \sum_{t=1}^{T} \left(\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\right)^\top \left(\boldsymbol{\theta}_t - \boldsymbol{\vartheta}_t\right) \leq \sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_{(t-1),*} \|\boldsymbol{\theta}_t - \boldsymbol{\vartheta}_t\|_{(t-1)} \tag{34}$$

$$\text{where} \quad \boldsymbol{\vartheta}_t = \arg\min_{\boldsymbol{\theta}\in\Theta} \left\{ q_{1:t-1}(\boldsymbol{\theta}) + \boldsymbol{\theta}^\top \sum_{\tau=1}^{t} \boldsymbol{\kappa}_\tau \right\} \tag{35}$$

is the prescient action that is selected with knowledge of next-round cost $\boldsymbol{\kappa}_t$ (instead of using predictions). Recalling the properties of the selected regularizer, we rewrite (34) as:

$$\mathcal{R}_T^{\theta} - q_{1:T-1}(\boldsymbol{\theta}^\star) \leq \sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2 \|\boldsymbol{\theta}_t - \boldsymbol{\vartheta}_t\|_2 \leq D_\Theta \sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2 \tag{36}$$

where we used the fact that Θ has a bounded diameter D_Θ. Combining (33) and (36), we can follow the rationale in [72, Sec. 7.6], and write:

$$\mathcal{R}_T^{\theta} - q_{1:T-1}(\boldsymbol{\theta}^\star) \leq \min \left\{ D_\Theta \sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2, \sum_{t=1}^{T} \frac{(1/\sigma)\|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}{\sqrt{\sum_{\tau=1}^{t-1} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}} \right\}$$

$$= \sum_{t=1}^{T} \min \left\{ D_\Theta \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2, \frac{(1/\sigma)\|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}{\sqrt{\sum_{\tau=1}^{t-1} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}} \right\}$$

$$= \sum_{t=1}^{T} \sqrt{\min \left\{ D_\Theta^2 \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2, \frac{(1/\sigma)^2\|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^4}{\sum_{\tau=1}^{t-1} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2} \right\}} \overset{(\alpha)}{\leq} \sum_{t=1}^{T} \sqrt{\frac{2}{\frac{1}{D_\Theta^2 \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2} + \frac{\sum_{\tau=1}^{t-1} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}{(1/\sigma)^2 \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^4}}}$$

$$= \sqrt{2} \sum_{t=1}^{T} \sqrt{\frac{(1/\sigma)^2 D_\Theta^2 \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^4}{(1/\sigma)^2 \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2 + D_\Theta^2 \sum_{\tau=1}^{t-1} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}} = \sum_{t=1}^{T} \frac{(1/\sigma) D_\Theta \sqrt{2} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}{\sqrt{(1/\sigma)^2 \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2 + D_\Theta^2 \sum_{\tau=1}^{t-1} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}}$$

$$\overset{(\beta)}{\leq} \sum_{t=1}^{T} \frac{(1/\sigma) D_\Theta \sqrt{2} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}{\sqrt{(1/\sigma)^2 \sum_{\tau=1}^{t} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}} = \sum_{t=1}^{T} \frac{D_\Theta \sqrt{2} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}{\sqrt{\sum_{\tau=1}^{t} \|\boldsymbol{\kappa}_\tau - \tilde{\boldsymbol{\kappa}}_\tau\|_2^2}} \overset{(\gamma)}{\leq} 2 D_\Theta \sqrt{2} \sqrt{\sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2}$$

where (α) uses that the minimum between two numbers is less than their harmonic mean; (β) assumes that $(1/\sigma) \leq D_\Theta$, which is satisfied by the proposed value for σ (see below); and (γ) applies an identify from [12, Lemma 3.5]. To conclude, it suffices to observe that $q_{1:T-1}(\boldsymbol{\theta}^\star)$ can be upper bounded due to boundedness of Θ as follows:

$$q_{1:T-1}(\boldsymbol{\theta}^\star) \leq \sigma D_\Theta^2 \sqrt{\sum_{t=1}^{T} \|\boldsymbol{\kappa}_t - \tilde{\boldsymbol{\kappa}}_t\|_2^2},$$

and the value of parameter σ that minimizes the constant factor above is $\sigma = 2\sqrt{2}/D_\Theta$.

8.2 Proof of Lemma 4.2

We start by characterizing the strong convexity of the entropic regularizer $r_{1:t}$ that we use in the primal update. We note that this is not the typical entropic regularizer used in FTRL (or Mirror Descent) algorithms, cf. [58]. Here, the regularizing parameter does not have a constant term (this allows us to get $O(1)$ for perfect predictions), and the constraint set is a set of simplices, i.e., a multi-simplex, instead of a single simplex binding all variables.

LEMMA 8.1. *Consider the convex set $\mathcal{X} = \{x_{ij} \geq 0 \ : \ \sum_{j\in\mathcal{J}} x_{ij} = 1, \ \ \forall i \in \mathcal{I}\}$, and the nonnegative convex function $r_{1:t} : \mathcal{X} \mapsto \mathbb{R}_+$ defined in* (18) *as:*

$$r_{1:t}(\boldsymbol{x}) = \frac{\eta_{1:t}}{2}\left(I \log J + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} x_{ij}\log x_{ij}\right), \quad \textit{where} \quad \eta_{1:t} = \eta\sqrt{\sum_{\tau=1}^{t}\|\boldsymbol{g}_\tau + \boldsymbol{w}_\tau - \tilde{\boldsymbol{g}}_\tau - \tilde{\boldsymbol{w}}_\tau\|_\infty^2}$$

Then, function $r_{1:t}(\boldsymbol{x})$ is 1-strongly convex with respect to the norm $\|\boldsymbol{x}\|_{(t)} = \|\boldsymbol{x}\|_1\sqrt{\frac{\eta_{1:t}}{I}}$.

PROOF. Let us define $\boldsymbol{x}_i = (x_{ij}, j \in \mathcal{J})$ and $\boldsymbol{y}_i = (y_{ij}, j \in \mathcal{J})$, and the unit simplex $\mathcal{X}_i = \{x_{ij} \geq 0 : \sum_{j\in\mathcal{J}} x_{ij} = 1\}$. Then, from the standard analysis of the entropic regularizer it holds that the (simpler) reguarlizer defined as

$$\hat{r}_i(\boldsymbol{x}_i) = \log J + \sum_{j\in\mathcal{J}} x_{ij}\log x_{ij} \tag{37}$$

is 1-strongly convex w.r.t. the ℓ_1 norm over $\mathcal{X}_i$, and therefore it holds:

$$\hat{r}_i(\boldsymbol{y}_i) \geq \hat{r}_i(\boldsymbol{x}_i) + \nabla\hat{r}_i(\boldsymbol{x}_i)^\top(\boldsymbol{y}_i - \boldsymbol{x}_i) + \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{x}_i\|_1^2, \quad \forall i \in \mathcal{I}, \tag{38}$$

Hence, we can write:

$$\begin{aligned}
\frac{r_{1:t}(\boldsymbol{y})}{\eta_{1:t}/2} = \sum_{i\in\mathcal{I}}\hat{r}_i(\boldsymbol{y}_i) &\geq \sum_{i\in\mathcal{I}}\hat{r}_i(\boldsymbol{x}_i) + \frac{\nabla r_{1:t}(\boldsymbol{x})}{\eta_{1:t}/2}^\top(\boldsymbol{y}-\boldsymbol{x}) + \sum_{i\in\mathcal{I}}\frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{x}_i\|_1^2 \\
&\overset{(\alpha)}{\geq} \sum_{i\in\mathcal{I}}\hat{r}_i(\boldsymbol{x}_i) + \frac{\nabla r_{1:t}(\boldsymbol{x})}{\eta_{1:t}/2}^\top(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2I}\|\boldsymbol{y}-\boldsymbol{x}\|_1^2 \\
&= \frac{r_{1:t}(\boldsymbol{x})}{\eta_{1:t}/2} + \frac{\nabla r_{1:t}(\boldsymbol{x})}{\eta_{1:t}/2}^\top(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2I}\|\boldsymbol{y}-\boldsymbol{x}\|_1^2 \Rightarrow \qquad (39) \\
r_{1:t}(\boldsymbol{y}) \geq r_{1:t}(\boldsymbol{x}) &+ \nabla r_{1:t}(\boldsymbol{x})^\top(\boldsymbol{y}-\boldsymbol{x}) + \frac{\eta_{1:t}}{I}\|\boldsymbol{y}-\boldsymbol{x}\|_1^2 \qquad (40)
\end{aligned}$$

where in (α) we used the inequality $(a_1 + a_2 + \ldots + a_n)^2 \leq n(a_1^2 + a_2^2 + \ldots + a_n^2)$. □

With this result at hand, we can proceed to prove Lemma 4.2 following a similar approach as in the proof of Lemma 4.1. From [66, Theorem 2], we can write:

$$\begin{aligned}
\mathcal{R}_T - r_{1:T-1}(\boldsymbol{x}^\star) &\leq \sum_{t=1}^{T}\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_{(t-1),*}^2 = \sum_{t=1}^{T}\frac{I}{\eta_{1:t-1}}\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2 \\
&= \sum_{t=1}^{T}\frac{I\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2}{\eta\sqrt{\sum_{k=1}^{t-1}\|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}}.
\end{aligned} \tag{41}$$

Similarly, from the proof of the same Theorem, we can write:

$$\mathcal{R}_T - r_{1:T-1}(\boldsymbol{x}^\star) \leq \sum_{t=1}^{T}\left(\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\right)^\top\left(\boldsymbol{x}_t - \boldsymbol{\chi}_t\right) \leq \sum_{t=1}^{T}\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_{(t-1),*}\|\boldsymbol{x}_t - \boldsymbol{\chi}_t\|_{(t-1)}$$

where $\quad \boldsymbol{\chi}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}}\left\{r_{1:t}(\boldsymbol{x}) + \boldsymbol{x}^\top\left(\boldsymbol{g}_{1:t+1} + \boldsymbol{w}_{1:t+1}\right)\right\}$

is the prescient action that is selected with knowledge of next-round cost $\boldsymbol{g}_{t+1}$. Recalling the properties of the entropic regularizer, we have:

$$\mathcal{R}_T - r_{1:T-1}(\boldsymbol{x}^\star) \le \sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_{(t-1),*} \|\boldsymbol{x}_t - \boldsymbol{\chi}_t\|_{(t-1)} = \sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty \|\boldsymbol{x}_t - \boldsymbol{\chi}_t\|_1 \Rightarrow$$

$$\mathcal{R}_T - r_{1:T-1}(\boldsymbol{x}^\star) \le \sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty \big(\|\boldsymbol{x}_t\|_1 + \|\boldsymbol{\chi}_t\|_1\big) \le 2I \sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty, \tag{42}$$

where in the last step we used the fact that $\boldsymbol{x}_t, \boldsymbol{\chi}_t \in \mathcal{X}$, and they have non-negative elements. Combining (41) and (42), we can follow the rationale in [72, Sec. 7.6], and write:

$$\mathcal{R}_T - r_{1:T-1}(\boldsymbol{x}^\star) \le \min\left\{ 2I \sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty, \sum_{t=1}^{T} \frac{I\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2}{\eta\sqrt{\sum_{k=1}^{t-1} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}} \right\}$$

$$= \sum_{t=1}^{T} \min\left\{ 2I\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty, \frac{I\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2}{\eta\sqrt{\sum_{k=1}^{t-1} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}} \right\}$$

$$= 2I \sum_{t=1}^{T} \min\left\{ \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty, \frac{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2}{2\eta\sqrt{\sum_{k=1}^{t-1} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}} \right\}$$

$$= 2I \sum_{t=1}^{T} \sqrt{\min\left\{ \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2, \frac{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^4}{4\eta^2\sum_{k=1}^{t-1} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2} \right\}}$$

$$\overset{(\alpha)}{\le} 2I \sum_{t=1}^{T} \sqrt{\frac{2}{\frac{1}{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2} + \frac{4\eta^2\sum_{k=1}^{t-1} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^4}}}$$

$$= 2\sqrt{2}I \sum_{t=1}^{T} \sqrt{\frac{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^4}{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2 + 4\eta^2\sum_{k=1}^{t-1} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}}$$

$$\overset{(\beta)}{\le} 2\sqrt{2}I \sum_{t=1}^{T} \sqrt{\frac{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^4}{4\eta^2\sum_{k=1}^{t} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}}$$

$$= \frac{\sqrt{2}I}{\eta} \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2}{\sqrt{\sum_{k=1}^{t} \|\boldsymbol{g}_k + \boldsymbol{w}_k - \tilde{\boldsymbol{g}}_k - \tilde{\boldsymbol{w}}_k\|_\infty^2}} \overset{(\gamma)}{\le} \frac{\sqrt{2}I}{\eta} \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2}$$

where (α) uses that the minimum between two numbers is less than their harmonic mean; (β) assumes that $\eta^2 \le 1/4$ or $\eta \le 1/2$; and in (γ) we applied the identity [12, Lemma 3.5]. To conclude, it suffices to observe that $r_{1:T-1}(\boldsymbol{x}^\star)$ can be upper bounded as follows:

$$r_{1:T-1}(\boldsymbol{x}^\star) = \frac{\eta_{1:T-1}}{2}\left(I \log J + \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} x_{ij} \log x_{ij} \right) \le \frac{\eta_{1:T-1}}{2} I \log J$$

$$= \frac{\eta I \log J}{2} \sqrt{\sum_{t=1}^{T-1} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2} \le \frac{\eta I \log J}{2} \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t + \boldsymbol{w}_t - \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{w}}_t\|_\infty^2}$$

8.3 Proof of Proposition 1

Iteration (12) requires the solution of a convex optimization problem. Since we use non-proximal regularizers, we can provide a closed-form expression using the KKT conditions [24, Chapter 4]. In detail, in order to calculate $\boldsymbol{\theta}_{t+1}$, we need to solve:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{\sigma_{1:t}}{2}\|\boldsymbol{\theta}\|^2 + \boldsymbol{\theta}^\top\left(\boldsymbol{\kappa}_{1:t} + \tilde{\boldsymbol{\kappa}}_{\boldsymbol{t+1}}\right) \\ \text{s.t.} \quad & \theta_i \geq \theta_i^l, \quad \forall i \in \mathcal{I}, \\ & \theta_i \leq \theta_i^u, \quad \forall i \in \mathcal{I}. \end{aligned}$$

where θ_i^l and θ_i^u are the lowest and largest values the dual variables can attain (and depend on the maximum utility values). First, we define the vectors $\boldsymbol{\omega}_t \doteq \boldsymbol{\kappa}_{1:t} + \tilde{\boldsymbol{\kappa}}_{t+1} \doteq (\omega_{it}, i \in \mathcal{I})$, $\boldsymbol{\theta}^l = (\theta_i^l, i \in \mathcal{I})$, $\boldsymbol{\theta}^u = (\theta_i^u, i \in \mathcal{I})$; and introduce the non-negative dual variables $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ to relax the respective constraints and define the Lagrangian:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{\sigma_{1:t}}{2}\|\boldsymbol{\theta}\|^2 + \boldsymbol{\theta}^\top\boldsymbol{\omega}_t + \boldsymbol{\lambda}^\top(\boldsymbol{\theta}^l - \boldsymbol{\theta}) + \boldsymbol{\mu}^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^u).$$

Applying the KKT conditions we can write for the optimal solution $\boldsymbol{\theta}^\star$, $\boldsymbol{\lambda}^\star$, and $\boldsymbol{\mu}^\star$:

(1) Stationarity:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{v}) = 0 \;\;\Rightarrow\;\; \sigma_{1:t}\theta_i^\star + \omega_{it} - \lambda_i^\star + \mu_i^\star = 0, \quad \forall i \in \mathcal{I}.$$

(2) Complementary slackness: $\lambda_i{}^\star(\theta_i^l - \theta_i^\star) = 0$, and $\mu_i{}^\star(\theta_i^\star - \theta_i^u) = 0$, $\forall i \in \mathcal{I}$.
(3) Primal feasibility: $\theta_i^l \leq \theta_i^\star \leq \theta_i^u$, $\forall i \in \mathcal{I}$.
(4) Dual feasibility: $\lambda_i^\star \geq 0, \mu_i^\star \geq 0$, $\forall i \in \mathcal{I}$.

Using the above conditions and exploring the different cases for satisfying the complementary slackness conditions, we can see from the proposed expression in (22), that indeed $\theta_{i,t+1}^\star$ can admit the following values:

- $\theta_{i,t+1}^\star = \frac{-\omega_{it}}{\sigma_{1:t}} \Rightarrow$ All 4 conditions are satisfied by setting $\lambda_i^\star = 0, \mu_i^\star = 0$.
- $\theta_{i,t+1}^\star = \theta_i^l \Rightarrow \theta_i^l \geq \frac{-\omega_{it}}{\sigma_{1:t}}$. Setting $\mu_i^\star = 0$, $\lambda_i^\star = \sigma_{1:t}\theta_i^l + \omega_{it} \geq 0$ satisfies all 4 conditions.
- $\theta_{i,t+1}^\star = \theta_i^u \Rightarrow \theta_i^u \leq \frac{-\omega_{it}}{\sigma_{1:t}}$. Setting $\lambda_i^\star = 0$, $\mu_i^\star = -\sigma_{1:t}\theta_i^u - \omega_{it} \geq 0$ satisfies all 4 conditions.

8.4 Proof of Proposition 2

The update (17) involves solving the convex problem (dropping the time index of variables):

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & r_{1:t}(\boldsymbol{x}) - \boldsymbol{x}^\top\left(\boldsymbol{g}_{1:t} + \boldsymbol{w}_{1:t} + \tilde{\boldsymbol{g}}_{t+1} + \tilde{\boldsymbol{w}}_{t+1}\right) \\ \text{s.t.} \quad & \sum_{j \in \mathcal{J}} x_{ij} = 1, \quad \forall i \in \mathcal{I}, \\ & x_{ij} \geq 0, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}. \end{aligned}$$

First, we define $\boldsymbol{\omega}_t \doteq \boldsymbol{g}_{1:t} + \boldsymbol{w}_{1:t} + \tilde{\boldsymbol{g}}_{t+1} + \tilde{\boldsymbol{w}}_{t+1} \doteq (\omega_{ijt}, i \in \mathcal{I}, j \in \mathcal{J})$ and introduce the dual variable vectors $\boldsymbol{\lambda} \in \mathbb{R}_+^{I \cdot J}$ and $\boldsymbol{\mu} \in \mathbb{R}^I$, to define the Lagrangian:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = r_{1:t}(\boldsymbol{x}) - \boldsymbol{x}^\top\boldsymbol{\omega}_t - \boldsymbol{\lambda}^\top\boldsymbol{x} + \sum_{i \in \mathcal{I}} \mu_i\Bigg(\sum_{j \in \mathcal{J}} x_{ij} - 1\Bigg).$$

The KKT conditions are:

(1) Stationarity: $\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \boldsymbol{0}$, which yields the following:

$$\begin{aligned}
&\text{if} \quad x_{ij} > 0 : \quad \frac{\eta_{1:t}}{2}\left(\log x_{ij} + 1\right) - \omega_{ijt} - \lambda_{ij} + \mu_i = 0, && \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},\\
&\text{if} \quad x_{ij} = 0 : \quad -\omega_{ijt} - \lambda_{ij} + \mu_i = 0, && \forall i \in \mathcal{I},\ \forall j \in \mathcal{J}.
\end{aligned}$$

(2) Complementary slackness: $\lambda_{ij}x_{ij} = 0, \forall i \in \mathcal{I}, j \in \mathcal{J}$.
(3) Primal feasibility: $x_{ij} \geq 0,\ \sum_{j\in\mathcal{J}} x_{ij} = 1,\ \forall i \in \mathcal{I}, \forall j \in \mathcal{J}$
(4) Dual feasibility: $\lambda_{ij} \geq 0, \forall i \in \mathcal{I}, j \in \mathcal{J}$.

Setting $\boldsymbol{\lambda} = \boldsymbol{0}$, and solving for μ_i in each equation, we obtain:

$$\mu_i = -\frac{\eta_{1:t}}{2}\left(\log x_{ij} + 1\right) + \omega_{ijt}, \quad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},$$

and replacing the proposed expression for $\boldsymbol{x}$ from (23), we get $\forall i \in \mathcal{I}$:

$$\mu_i = -\frac{\eta_{1:t}}{2}\left(\frac{2\omega_{ijt}}{\eta_{1:t}} - \log\left(\sum_{j\in\mathcal{J}} \exp\left(\frac{2\omega_{ijt}}{\eta_{1:t}}\right)\right) + 1\right) + \omega_{ijt} = \frac{\eta_{1:t}}{2}\log\left(\sum_{j\in\mathcal{J}} \exp\left(\frac{2\omega_{ijt}}{\eta_{1:t}}\right)\right) - \frac{\eta_{1:t}}{2},$$

where notice that μ_i contains summations over all elements of $\mathcal{J}$, and hence its value does not depend on the variable derivative j-wise. Therefore, this solution satisfies all KKT conditions, since the primal variables and the λ_{ij} variables are nonegative, and it holds:

$$\sum_{j\in\mathcal{J}} \frac{\exp\left(2\omega_{ijt}/\eta_{1:t}\right)}{\sum_{j\in\mathcal{J}}\exp\left(2\omega_{ijt}/\eta_{1:t}\right)} = 1, \quad \forall i \in \mathcal{I},\ j \in \mathcal{J},$$

8.5 Proof of Theorem 5.1

Using [81, Lemma 2], we can write:

$$F_\alpha\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{u}_t(\boldsymbol{y}_t)\right) = \min_{\boldsymbol{\theta}\in\Theta}\left\{(-F_\alpha)^\star(\boldsymbol{\theta}) - \boldsymbol{\theta}\cdot\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{u}_t(\boldsymbol{y}_t)\right\} = \min_{\boldsymbol{\theta}\in\Theta}\left\{\frac{1}{T}\sum_{t=1}^{T}(-F_\alpha)^\star(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top\boldsymbol{u}_t(\boldsymbol{y}_t)\right\}.$$

Following the definition of $G_\alpha(\{\boldsymbol{y}_t\}_t)$ and combining it with the above result, we can write:

$$\begin{aligned}
G_\alpha(\{\boldsymbol{y}_t\}_t) &= F_\alpha\left(\frac{1}{T}\sum_{t\in\mathcal{T}} u_t(\boldsymbol{y}_t)\right) - \frac{1}{T}\sum_{t\in\mathcal{T}} c_t(\boldsymbol{y}_t) = \min_{\boldsymbol{\theta}\in\Theta}\left\{\frac{1}{T}\left[\sum_{t=1}^{T}(-F_\alpha)^\star(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top\boldsymbol{u}_t(\boldsymbol{y}_t)\right]\right\} - \frac{1}{T}\sum_{t=1}^{T} c_t(\boldsymbol{y}_t)\\
&= \min_{\boldsymbol{\theta}\in\Theta}\left\{\frac{1}{T}\left[\sum_{t=1}^{T}(-F_\alpha)^\star(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top\boldsymbol{u}_t(\boldsymbol{y}_t) - c_t(\boldsymbol{y}_t)\right]\right\} = \min_{\boldsymbol{\theta}\in\Theta}\left\{\frac{1}{T}\sum_{t=1}^{T}\Psi_t^c(\boldsymbol{\theta}, \boldsymbol{y}_t)\right\}. \qquad (43)
\end{aligned}$$

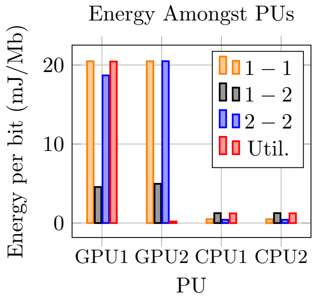


Fig. 11. Energy dispersion amongst PUs for different $\alpha - \beta$ pairs. *Util.* stands for utilitarian algorithm.

Denoting with $\mathcal{R}_T^{y,c}$ the primal-space regret (in analogy with (10)), we have:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\Psi_t^c(\boldsymbol{\theta}_t,\boldsymbol{y}_t)+\frac{\mathcal{R}_T^{y,c}}{T}=\frac{1}{T}\sum_{t=1}^{T}\Psi_t^c(\boldsymbol{\theta}_t,\boldsymbol{y}_\star)=\frac{1}{T}\sum_{t=1}^{T}(-F_\alpha)^\star(\boldsymbol{\theta}_t)-\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_t^\top\boldsymbol{u}_t(\boldsymbol{y}^\star)-\frac{1}{T}\sum_{t=1}^{T}c_t(\boldsymbol{y}^\star)\\
&\overset{(\gamma_1)}{\geq}(-F_\alpha)^\star\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_t\right)-\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_t^\top\boldsymbol{u}_t(\boldsymbol{y}^\star)-\frac{1}{T}\sum_{t=1}^{T}c_t(\boldsymbol{y}^\star)\\
&=(-F_\alpha)^\star(\bar{\boldsymbol{\theta}})-\bar{\boldsymbol{\theta}}\cdot\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{u}_t(\boldsymbol{y}^\star)\right)-\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t-\bar{\boldsymbol{\theta}})^\top\cdot\boldsymbol{u}_t(\boldsymbol{y}^\star)-\frac{1}{T}\sum_{t=1}^{T}c_t(\boldsymbol{y}^\star)\\
&\geq\min_{\boldsymbol{\theta}\in\Theta}\left\{\frac{1}{T}\sum_{t=1}^{T}(-F_\alpha)^\star(\boldsymbol{\theta})-\boldsymbol{\theta}^\top\boldsymbol{u}_t(\boldsymbol{y}^\star)-c_t(\boldsymbol{y}^\star)\right\}-\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t-\bar{\boldsymbol{\theta}})^\top\boldsymbol{u}_t(\boldsymbol{y}^\star)\\
&=\min_{\boldsymbol{\theta}\in\Theta}\left\{\frac{1}{T}\sum_{t=1}^{T}\Psi_t^c(\boldsymbol{\theta},\boldsymbol{y}^\star)\right\}-\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t-\bar{\boldsymbol{\theta}})^\top\boldsymbol{u}_t(\boldsymbol{y}^\star)\overset{(\gamma_2)}{=}G_\alpha(\boldsymbol{y}^\star)-\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t-\bar{\boldsymbol{\theta}})^\top\boldsymbol{u}_t(\boldsymbol{y}^\star),\quad(44)
\end{aligned}$$

where (γ_1) follows from Jensen's inequality and the convexity of $(-F_\alpha)^\star$, and in (γ_2) we used (43). Next, we define the dual-space regret $\mathcal{R}_T^{\theta,c}$ (in analogy to (11)) and relate it to function G_α, namely:

$$\begin{aligned}
\mathcal{R}_T^{\theta,c}&=\sum_{t=1}^{T}\Psi_t^c(\boldsymbol{\theta}_t,\boldsymbol{y}_t)-\sum_{t=1}^{T}\Psi_t^c(\boldsymbol{\theta},\boldsymbol{y}_t)\quad\text{for every }\boldsymbol{\theta}\in\Theta\\
&\overset{(\gamma_3)}{=}\sum_{t=1}^{T}\Psi_{\alpha,t}^c(\boldsymbol{\theta}_t,\boldsymbol{y}_t)-TG_\alpha([\boldsymbol{y}_t]).\qquad(45)
\end{aligned}$$

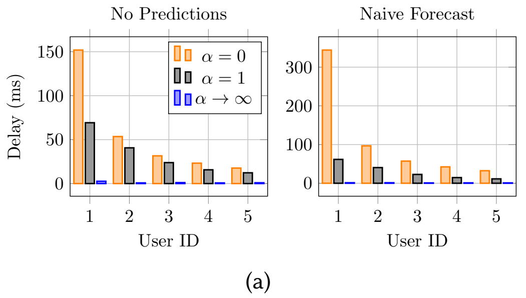
where (γ_3) follows from the definition of $G_\alpha(\{\boldsymbol{y}_t\}_t)$ as the minimizer of the averaged proxy function values w.r.t. $\boldsymbol{\theta}\in\Theta$, see (43). Now, we can combine (44) and (45), and write:

$$G_\alpha([\boldsymbol{y}_t])+\frac{\mathcal{R}_T^{\theta,c}}{T}=\frac{1}{T}\sum_{t=1}^{T}\Psi_t^c(\boldsymbol{\theta}_t,\boldsymbol{y}_t)\geq G_\alpha(\boldsymbol{y}^\star)-\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{\theta}_t-\bar{\boldsymbol{\theta}})^\top\boldsymbol{u}_t(\boldsymbol{y}^\star)-\frac{\mathcal{R}_T^{y,c}}{T}.$$

Rearranging, we arrive at the main result of the theorem.

8.6 Additional Experiments and Evaluation Results

This section includes further results that could not be included in the main part of the paper due to lack of space. All the results presented in this section are obtained using the O-RAN compliant experimental platform presented in Sec. 6.



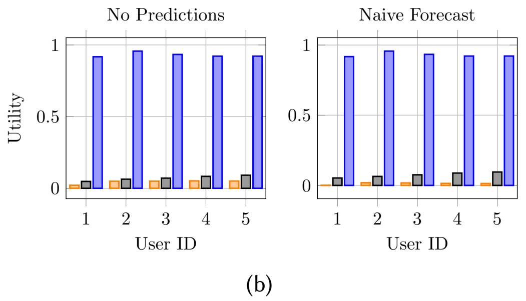


Fig. 12. **(a):** Delay dispersion amongst users for different α, non-optimistic FTRL (left) and OFTRL with Naive forecast (right). **(b):** Utility (probability of empty buffer) dispersion amongst users for different α values, non-optimistic FTRL (left) and OFTRL with Naive forecast (right).

We start with the assignment (compute control) policy. Fig. 11 shows the dispersion of throughput amongst vBSs and energy of PUs using the horizon fair, and utilitarian algorithms as in Sec. 6.1. In Fig. 11, the difference between $\alpha = 1, \beta = 1$ (orange) and $\alpha = 1, \beta = 2$ (black) indicates that as β is increased, energy is distributed more fairly among servers with the horizon fair algorithm. Also note that the horizon fair algorithm disperses the energy fairly and uses both of the GPUs, whereas the utilitarian algorithm chooses to use only the faster and cheaper GPU to reduce its energy with an unfair use.

Next, we provide additional results on the radio control policy (`minTB`). Fig. 12 shows the dispersion of actual measured delay, and percentage of the user buffer being empty, amongst different users when using the non-optimistic FTRL and OFTRL with Naive forecast [47] algorithms. We consider the configuration of Scenario 2, detailed in Sec. 6.2. We see that the delays and utilities of the users are dispersed more fairly as α increases.

8.7 Derivation of Convex Utility and Cost Functions in Sec. 6.2

Our policy makes decisions every 100 ms, and we need to approximate the probability of empty buffer and expected energy cost between decisions, depending on $\boldsymbol{y}_t, \boldsymbol{b}_t, \boldsymbol{\rho}_t$, and $\boldsymbol{s}_t$. To approximate $\boldsymbol{u}_t$, we assume that data generation of each user b_{it} follows a Poisson distribution, where the times between data generations are exponentially distributed with the parameter $1/b_{it}$ and each data generation consists of ρ_{it} number of bits. We stress, however, that this is a non-binding assumption (other models can be studied), and that we allow the parameters of the distribution to change arbitrarily (based on the adversary model) across the different slots.

We designed the system such that the user data is transmitted when the number of bits in the buffer or each user $i \in \mathcal{I}$ exceeds the threshold y_{it}, where we denote the number of bits as $B_{it} = b_{it}\rho_{it}$. Here, for notational convenience we drop the subscripts i and t and derive a utility function for each user i in each time slot t. Additionally, we define a new time variable $\tau \in [0, 1)$ within the time slot t and denote the number of bits in the user buffer at time τ as B_τ. Next, we calculate $\mathbf{Pr}\left(B_\tau > 0 \text{ and } \tau < \frac{y}{b\rho}\right)$ as:

$$\mathbf{Pr}\left(B_\tau > 0 \text{ and } \tau < \frac{y}{b\rho}\right) = \mathbf{Pr}\left(\tau \geq \text{time of the first bit generation and } \tau < \frac{y}{b\rho}\right)$$
$$= \mathbf{Pr}\left(\text{time of the first bit generation} \leq \tau < \frac{y}{b\rho}\right) = \int_0^{\frac{y}{b\rho}} \left(1 - e^{-b\tau}\right) d\tau$$

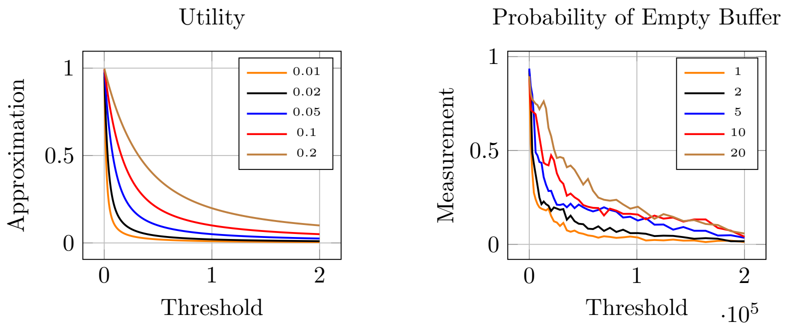


Fig. 13. Comparison of $u(y)$ with the real measurement of probability of empty buffer. Legends are ρ (left) and traffic multiplier (right).

Now, we calculate the portion of time the user has non-empty buffer when $\tau \in [0, 1)$ as

$$1 - u(y) = \mathbf{Pr}\left(B_\tau > 0 \text{ and } \tau < \frac{y}{b\rho}\right)\frac{b\rho}{y} = \frac{\rho e^{-\frac{y}{\rho}} + y - \rho}{y}$$

Thus, we have $u(y) = \frac{\rho}{y}\left(1 - e^{-\frac{y}{\rho}}\right)$. Note that when $y \to 0, u(y) = 1$ and for larger y values $u(y) \approx \frac{\rho}{y}$. We approximate the cost function similarly, since the probability of empty buffer is an indicator of the rate of data transmissions, i.e., the buffer is empty right after the transmission up until the first data generation after the last transmission. Therefore, we can approximate the number of data transmissions by $b \cdot u(y)$. We then multiply this with the cost multiplier due to SNR, and the cost scaling parameter to calculate hardware cost induced by user i as $c(y) = \varphi\beta(s)b\frac{\rho}{y}\left(1 - e^{-\frac{y}{\rho}}\right)$. We sum this cost function for all users to calculate the HA cost. We show that the functions are convex as $u''(y) \geq 0$ is always satisfied.

Fig. 13 demonstrates a comparison between our approximation function $u(y)$ and the real measurement of empty buffer probability gathered using our testbed. Here, in the approximation we modify ρ, and in the real measurements we multiply the user traffics to increase the demand.

8.8 Convexity of Functions in Sec. 6.1

First, we prove that the utility function in Sec. 6.1 is indeed concave.

$$u_{ij}(\boldsymbol{x}) = x_{ij}\lambda_i \cdot \min\left\{1,\ 1 - \frac{1}{C_j}\left(\sum_{k\in\mathcal{I}} \frac{x_{kj}\lambda_k}{n_k}(\zeta_k^j n_k + o_k^j) - C_j\right)\right\}$$

We do not use the time subscript t for notational simplicity. Note that the piecewise minimum of two concave functions is also concave, and it is sufficient to prove that both functions inside min{} after multiplied with $x_{ij}\lambda_i$ are concave. The LHS, $x_{ij}\lambda_i$ is a linear function, thus concave. Therefore, it is sufficient to show that:

$$x_{ij}\lambda_i - \frac{x_{ij}\lambda_i}{C_j}\left(\sum_{k\in\mathcal{I}} \frac{x_{kj}\lambda_k}{n_k}(\zeta_k^j n_k + o_k^j) - C_j\right),$$

is concave. We calculate the Hessian matrix $\boldsymbol{H}$ of

$$f_{ij}(\boldsymbol{x}) = x_{ij}\lambda_i - \frac{x_{ij}\lambda_i}{C_j}\left(\sum_{k\in\mathcal{I}} \frac{x_{kj}\lambda_k}{n_k}\left(\zeta_k^j n_k + o_k^j\right) - C_j\right),$$

as $\boldsymbol{H} \doteq \nabla \boldsymbol{g}$ where $\boldsymbol{g} \doteq \nabla f_{ij}(\boldsymbol{x})$, $\boldsymbol{H} \in \mathbb{R}^{(I \cdot J) \times (I \cdot J)}$ and $\boldsymbol{g} \in \mathbb{R}^{(I \cdot J)}$. Denoting $\frac{\lambda_i}{n_i}\left(\zeta_i^j n_i + o_i^j\right) \doteq k_i$, we can write:

$$\begin{aligned}
g_{ij} &= 2\lambda_i - x_{ij}\frac{2\lambda_i k_i}{C_j} - \frac{\lambda_i}{C_j}\sum_{i' \neq i} x_{i'j} k_{i'}, \\
g_{i'j} &= -\frac{x_{ij}\lambda_i k_{i'}}{C_j}, \\
g_{ij'} &= 0, \\
g_{i'j'} &= 0,
\end{aligned} \tag{46}$$

for the values of $\boldsymbol{g}$ where $i' \neq i, j' \neq j$. The Hessian matrix $\boldsymbol{H} = \nabla \boldsymbol{g}$ has the following elements:

$$\begin{aligned}
&H_{ij,ij} = -\frac{2\lambda_i k_i}{C_j}, && H_{ij,i'j} = -\frac{\lambda_i k_{i'}}{C_j}, && H_{ij,ij'} = 0, && H_{ij,i'j'} = 0, \\
&H_{i'j,ij} = -\frac{\lambda_i k_{i'}}{C_j}, && H_{i'j,i'j} = 0, && H_{i'j,ij'} = 0, && H_{i'j,i'j'} = 0, \\
&H_{ij',ij} = 0, && H_{ij',i'j} = 0, && H_{ij',ij'} = 0, && H_{ij',i'j'} = 0, \\
&H_{i'j',ij} = 0, && H_{i'j',i'j} = 0, && H_{i'j',ij'} = 0, && H_{i'j',i'j'} = 0.
\end{aligned} \tag{47}$$

Hence, $\boldsymbol{H}$ is a negative semi definite matrix, thus $f_{ij}(\boldsymbol{x})$ is a concave function of $\boldsymbol{x}$.

The cost efficiency function $h_j(\boldsymbol{x}_j)$ is a linear function of $\boldsymbol{x}$, hence concave.

Received October 2023; revised January 2024; accepted January 2024