Dynamic Edge Computing empowered by Reconfigurable Intelligent Surfaces

Paolo Di Lorenzo, Mattia Merluzzi, Emilio Calvanese Strinati, Sergio Barbarossa

Di Lorenzo *et al. J Wireless Com Network* (2022) 2022:122
https://doi.org/10.1186/s13638-022-02203-6

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Dynamic edge computing empowered by reconfigurable intelligent surfaces

Paolo Di Lorenzo[1,2*], Mattia Merluzzi[3], Emilio Calvanese Strinati[3] and Sergio Barbarossa[1,2]

*Correspondence: paolo.dilorenzo@uniroma1.it

[1] Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, Rome, Italy
[2] Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Parma, Italy
[3] CEA Leti, University Grenoble Alpes, 38000 Grenoble, France

Abstract

In this paper, we propose a novel algorithm for energy-efficient low-latency dynamic mobile edge computing (MEC), in the context of beyond 5G networks endowed with reconfigurable intelligent surfaces (RISs). We consider a scenario where new computing requests are continuously generated by a set of devices and are handled through a dynamic queueing system. Building on stochastic optimization tools, we devise a dynamic learning algorithm that jointly optimizes the allocation of radio resources (i.e., power, transmission rates, sleep mode and duty cycle), computation resources (i.e., CPU cycles), and RIS reflectivity parameters (i.e., phase shifts), while guaranteeing a target performance in terms of average end-to-end delay. The proposed strategy enables dynamic control of the system, performing a low-complexity optimization on a per-slot basis while dealing with time-varying radio channels and task arrivals, whose statistics are unknown. The presence and optimization of RISs helps boosting the performance of dynamic MEC, thanks to the capability to shape and adapt the wireless propagation environment. Numerical results assess the performance in terms of service delay, learning, and adaptation capabilities of the proposed strategy for RIS-empowered MEC.

Keywords: Mobile edge computing, Reconfigurable intelligent surfaces, Lyapunov stochastic optimization, Dynamic resource allocation

1 Introduction

With the advent of beyond 5G networks [1, 2], mobile communication systems are evolving from a pure communication framework to enablers of a plethora of new services (including *verticals*), such as Industry 4.0, Internet of Things (IoT), and autonomous driving, building on the tight integration of communication, computation, caching, and control [3–5]. These new services have very different requirements, and they generally involve massive data processing within low end-to-end delays. Among several technology enablers at different layers (e.g., artificial intelligence, network function virtualization, millimeter-wave communications), a prominent role will be played by mobile edge computing (MEC), whose aim is to move cloud functionalities (e.g., computing and storage resources) at the edge of the wireless network to avoid the relatively long and highly variable delays necessary to reach centralized clouds. MEC-enabled networks allow user equipment (UE) to offload computational tasks to nearby processing units or edge servers (ESs), typically placed close to access points (APs), in order to run the computation

on the UE' behalf. However, since ESs have much smaller computation capabilities than the servers in the cloud, the available resources (i.e., radio, computation, energy) have to be properly managed to provide the end users with a satisfactory Quality of Service (QoS). In particular, since the end-to-end delay includes a communication time and a computation time, the resources available at the wireless network edge must be managed jointly, learning over time the best joint resource allocation in a dynamic and data-driven fashion.

Related works on MEC There is a wide literature on computation offloading, aimed at jointly optimizing communication and computation resources in both static and dynamic MEC scenarios [6–13]. Recent surveys on the topic appear also in [14, 15]. A possible classification of computation offloading problems is between *static* and *dynamic* strategies. The static formulation deals with short time applications, in which mobile users send a single computation request, typically specifying also a service time [6, 7, 16, 17]. Conversely, in a dynamic scenario, the application continuously generates data to be processed, sometimes with an unknown rate. A typical example could be the transmission of a video, recorded by a mobile device, to be processed at the ES side for pattern recognition or anomaly detection. The dynamic formulation is also useful to handle users' mobility, which is a central problem in mobile networks and becomes even more central in a MEC environment, where mobility may require handover mechanisms involving both radio access points (APs) and ESs. In [18], a dynamic formulation is proposed, with a strategy based on Lyapunov optimization in a cloud computing framework. In [11], a Lyapunov based strategy is proposed, for the joint optimization of radio and computation resources, to minimize the users' energy consumption under E2E delay constraints. In [10], the authors investigate a scenario with multiple APs and edge servers, where an assignment strategy based on matching theory is proposed, coupled with the tools of Lyapunov optimization and Extreme Value Theory to control reliability. In [13], a discontinuous mobile edge computing framework is proposed to minimize the energy consumption under latency constraints, considering a holistic approach that comprises UE, APs, and ESs. In [19], a deep reinforcement learning strategy driven by Lyapunov optimization is proposed to enable stable offloading in dynamic MEC, while the work in [20] proposed a multi-task learning based feedforward neural network trained to jointly optimize the offloading decision and computational resource allocation. Finally, in [5], a dynamic resource allocation framework is proposed for edge learning, encompassing communication, computation, and inference/training aspects of the learning task.

RIS-empowered wireless networks All the aforementioned works considered the presence of a suitable wireless propagation environment to enable edge computing. However, as highlighted in [21], moving toward millimeter-wave (mmWave) communications (and beyond), poor channel conditions due to mobility, dynamic environments, and blocking events, might severely hinder the performance of MEC systems. More explicitly, the devices at the cell edge or those affected by blockages usually suffer from low offloading rates, which increase both the latency and energy consumption of computation offloading. This fact prevents current MEC systems from guaranteeing the strict latency and reliability constraints required by several envisioned applications (e.g., smart factory, autonomous driving, etc.). Also, from the computational point of view, a lower

offloading probability implies that a large fraction of the computing resources available at the ESs remain idle due to the limited volume of received tasks, thus leading to an under-exploitation of the capabilities of the wireless edge. Therefore, it is of fundamental importance to enhance the efficiency of MEC systems by empowering their wireless offloading links.

In this context, a strong performance boost can be achieved with the advent of reconfigurable intelligent surfaces [22–24], which are artificial surfaces made of hundreds of nearly passive (sometimes, also active) reflective elements that can be programmed and controlled to realize dynamic transformations of the wireless propagation environment, in both indoor and outdoor scenarios. More precisely, an RIS is an array of backscatterers, where each element applies an individual phase-shift (and/or an amplitude and/or a polarization rotation) with which it backscatters an incident wave [25–27], with the aim of creating a controllable reflected beam. RISs enable full programmability of the wireless propagation environment and dynamically create service *boosted areas* where capacity, energy efficiency, and reliability can be dynamically traded to meet momentary and location dependent requirements [24]. RISs offer new opportunities to boost uplink and downlink capacities and to counteract blocking effects in case of directive mmWave communications.

In the literature, several works have already investigated the optimization of RIS-empowered wireless communications. In [28], a joint transmit power allocation and phase shift design was developed for an RIS-based multi-user system to maximize the energy efficiency. In [29], the authors considered a downlink RIS-assisted multi-user communication system and studied a joint transmission and reflection beamforming problem to minimize the total transmit power. Also, very recently, a few papers exploited RISs to enhance the performance of MEC systems, considering both edge caching [30] and static computation offloading [31–34]. A very nice overview paper on RIS-empowered MEC systems can also be found in [21]. In particular, in [31], the authors propose a latency-minimization problem for multi-device scenarios, which optimizes the computation offloading volume, the edge computing resource allocation, the multi-user detection matrix, and the RIS phase shifts, subject to a total edge computing capability. Reference [32] maximizes instead the number of processed bits for computation offloading, optimally designing the ES CPU frequency, the offloading time allocation, the transmit power of each device, and phase shifts of the RIS. Then, the work in [33] exploits RISs to maximize the performance of a machine learning task run at the edge server, acting jointly on radio parameters such as power of UE, beamforming vectors of AP, and RIS parameters. Finally, reference [34] proposes optimization-based and data-driven solutions for RIS-empowered multi-user mobile edge computing, maximizing the total completed task-input bits of all UE with limited energy budgets. All these previous works focus on a *static* edge computing scenario. Conversely, in this paper we focus on a *dynamic* scenario, where UE continuously generate data to be offloaded (e.g., a video stream for object detection), experiencing time varying context parameters (e.g., wireless channels conditions, data generation, server utilization, etc.).

Contributions of the paper In this work, we propose a novel algorithmic framework for energy-efficient, RIS-empowered dynamic computation offloading in mobile edge computing systems. To the best of our knowledge, this is the first contribution available in

the literature in the context of dynamic MEC empowered by RISs, and extends the preliminary conference precursor in [35]. In our dynamic system model, at each time slot, new offloading requests are generated by the UE, and are handled through a dynamic queueing system that accounts for both communication (i.e., uplink and downlink) and processing delays. In view of this dynamic MEC scenario, we devise a dynamic algorithm that learns over time the optimal radio parameters (i.e., powers, rates) for both UE and AP (including also AP sleep mode and duty cycle), computation resources (i.e., CPU cycles) of the ES, and RIS reflectivity parameters (i.e., phase shifts), with the aim of enabling energy-efficient mobile edge computing with low end-to-end latency guarantees. The method hinges on Lyapunov stochastic optimization, and allocates resources in a dynamic fashion requiring only low-complexity operations at each slot (with semi-closed form expressions). Furthermore, the method does not require any prior knowledge of channel and data arrival statistics, and is able to learn and adapt in real-time to changes in the environment due to, e.g., mobility of UE's or channel blocking. In this way, Lyapunov stochastic optimization acts as a method that dynamically learns optimal control policies (i.e., resource allocation) over time in an online and data-driven fashion (i.e., based on the online observation of channel states and task-related parameters.) To summarize, the main contribution of this paper is threefold: (i) A novel system model of the *dynamic* MEC problem involving the presence of multiple RISs, which controls system latency thanks to the introduction of proper communication and computation queues; (ii) a novel long-term problem formulation aimed at exploring the trade-off between energy and latency of dynamic computation offloading on average sense; (iii) an adaptive algorithmic solution based on Lyapunov optimization capable to track and control on the fly the dynamic behavior of the system. To the best of our knowledge, this is a novel approach that cannot be found in any of the previous works on RIS-empowered MEC such as, e.g., [31–34]. Finally, we assess the performance of the proposed strategy through numerical simulations, illustrating how RISs help boosting the performance of MEC systems.

2 System model

We consider a scenario with K edge devices, an access point (AP) equipped with an edge server (ES), and I RISs, as illustrated in Fig. 1. Time is divided in slots indexed by t and of equal duration τ_l. We consider a block-fading model where the wireless channel is assumed to be static within each slot, whose duration τ_l is designed with respect to the channel coherence time. Also, the overall slot duration τ_l is divided into two portions: a period of τ_s seconds dedicated to control signaling, and a period of τ seconds for the actual three phases of computation offloading (i.e., uplink, processing at the ES, and downlink). Here, we assume that control signaling happens before computation offloading due to the need of exchanging the state variables necessary to run the optimization algorithm and allocate radio and computation resources. The quantification of τ and τ_s depends on typical trade-offs between complexity and performance: a more accurate optimization could require a longer τ_s, thus leaving less time for transmission and computation, and vice versa.

We assume that the direct link between the users and the AP can be possibly impaired by the presence of obstacles, which attenuate or eventually block the communication, as

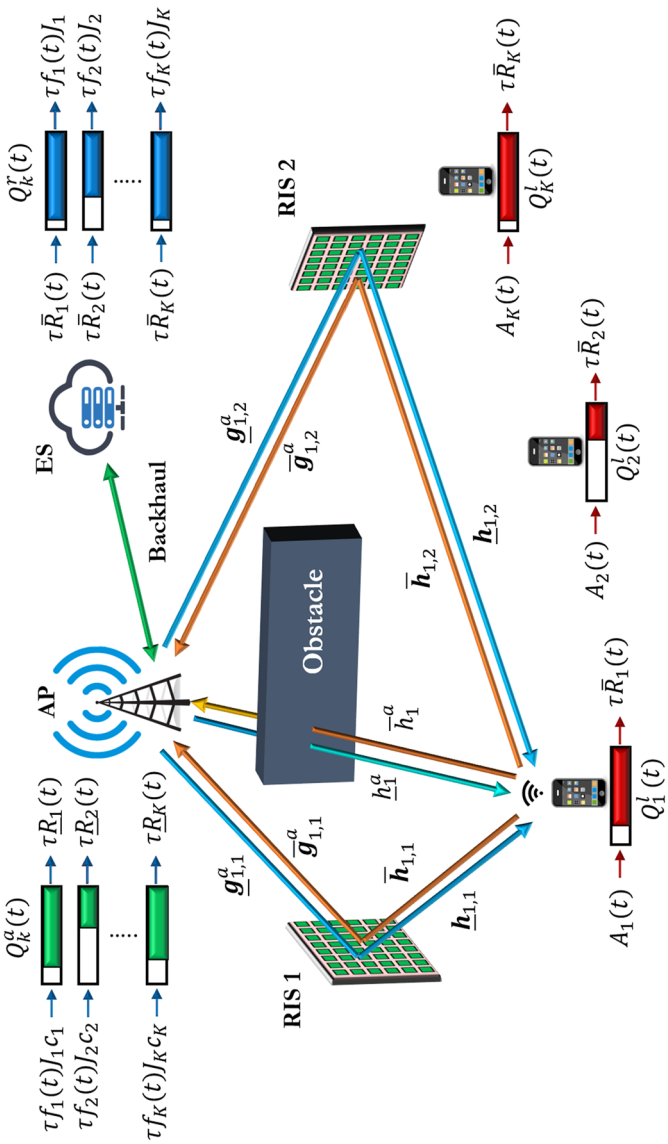


Fig. 1 RIS-empowered dynamic edge computing scenario

shown in Fig. 1. The presence of the RISs helps counteract this detrimental effect by allowing alternative communication paths. However, also in the case without obstacles, the RISs typically enhance performance [22]. We assume that synchronization is enforced by the AP, which also controls the behavior of the RISs. Furthermore, as typically done in cellular systems, the AP assigns orthogonal channels to its connected UE, thus preventing interference among them. Multi-cell scenarios for RIS-empowered distributed edge computing with inter-cell interference will be addressed in future investigations. In the sequel, we present the mathematical model of our dynamic system, considering RIS-enhanced communications, queueing model, and energy consumption.

2.1 RIS-enhanced communications

We consider a MEC system endowed with I (nearly) passive RISs, where the i-th RIS is composed of N_i reflecting elements. The RIS i at time t is described by the reflectivity matrix:

$$\mathbf{\Phi}_i(t) = \text{diag}\{m_{i,1}(t)e^{j\phi_{i,1}(t)}, \ldots, m_{i,N_i}(t)e^{j\phi_{i,N_i}(t)}\}, \tag{1}$$

for all i, t, where $m_{i,l}(t) \in \{0,1\}$ (i.e., the l-th reflective element of RIS i is active or not at time t), and $\phi_{i,l}(t) \in \left\{\frac{2k\pi}{2^{b_i}}\right\}_{k=0}^{2^{b_i}-1}$ (i.e., the phases are quantized using b_i bits) [26]. Equivalently, letting $v_{i,l}(t) = m_{i,l}(t)e^{j\phi_{i,l}(t)}$, we have $\mathbf{\Phi}_i(t) = \text{diag}\{\boldsymbol{v}_i(t)\}$, $\forall i,t$, where $\boldsymbol{v}_i(t) = \{v_{i,l}(t)\}_{l=1}^{N_i}$, with

$$v_{i,l}(t) \in \mathcal{S}_i = \left[0, \left\{e^{j\frac{2m\pi}{2^{b_i}}}\right\}_{m=0}^{2^{b_i}-1}\right], \quad \forall i,l,t. \tag{2}$$

Also, let $\boldsymbol{v}(t) = \{\boldsymbol{v}_i(t)\}_{i=1}^{I}$. In the sequel, we will use the overline notation for uplink parameters, and the underline notation for downlink. We consider an AP endowed with N_a antennas, thus leading to single input multiple output (SIMO) uplink communications, and multiple input single output (MISO) downlink communications. We assume that the AP performs analog beamforming through a vector $\mathbf{w}(t) \in \mathbb{C}^{N_a \times 1}$ that can be selected from a codebook $\mathcal{C}$ of available combiners/precoders, such that $\|\mathbf{w}(t)\|^2 = 1$ for all $\mathbf{w}(t) \in \mathcal{C}$. Also, we assume that the same beamforming vector $\mathbf{w}(t)$ is used for both uplink and downlink phase at time t. In this setting, the SIMO uplink transmission rate between user k and the AP reads as:

$$\overline{R}_k(t) = \overline{B}_k \log_2\left(1 + \overline{\alpha}_k(\boldsymbol{v}(t), \mathbf{w}(t))\overline{p}_k(t)\right) \qquad \text{for all } k = 1, \ldots, K, \tag{3}$$

where $\overline{p}_k(t)$ denotes the power transmitted by user k at time t, and

$$\overline{\alpha}_k(\boldsymbol{v}(t), \mathbf{w}(t)) = \frac{\left|\mathbf{w}(t)^H\left(\overline{\boldsymbol{h}}_k^a(t) + \sum_{i=1}^{I}\overline{\boldsymbol{G}}_{k,i}^a(t)\,\text{diag}(\boldsymbol{v}_\text{i}(\text{t}))\,\overline{\boldsymbol{h}}_{\text{k,i}}(\text{t})\right)\right|^2}{N_0\overline{B}_k} \tag{4}$$

is the RIS-dependent normalized uplink channel coefficient, the superscript ^{H} denotes the Hermitian operator, and $\overline{\boldsymbol{h}}_k^a(t) \in \mathbb{C}^{N_a \times 1}$ represents the direct uplink channel coefficient between user k and the AP; whereas, $\overline{\boldsymbol{h}}_{k,i}(t) \in \mathbb{C}^{N_i \times 1}$ and $\overline{\boldsymbol{G}}_{k,i}^a(t) \in \mathbb{C}^{N_a \times N_i}$ contain all the uplink channel coefficients between user k and RIS elements, and between RIS

elements and the AP, respectively. Specific models for $\overline{\boldsymbol{h}}_k^a(t)$, $\overline{\boldsymbol{h}}_{k,i}(t)$ and $\overline{\boldsymbol{G}}_{k,i}^a(t)$ can be found in [36]. Furthermore, $\overline{B}_k$ denotes the bandwidth allocated to user k for the uplink, and N_0 is the receiver noise power spectral density.

In our system, we assume that uplink and downlink happen simultaneously, using a frequency division duplexing scheme. Thus, similarly to (3), the downlink transmission rate between the AP and user k reads as:

$$\underline{R}_k(t) = \underline{B}_k \log_2\left(1 + \underline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t))\underline{p}_k(t)\right) \qquad \text{for all } k = 1, \ldots, K, \tag{5}$$

where $\underline{p}_k(t)$ denotes the power transmitted by the AP toward user k at time t, and

$$\underline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t)) = \frac{\left|\left(\underline{\boldsymbol{h}}_k^a(t) + \sum_{i=1}^{I} \underline{\boldsymbol{h}}_{k,i}(t)\,\mathrm{diag}(\boldsymbol{\nu}_\mathrm{i}(\mathrm{t}))\,\underline{\boldsymbol{G}}_{\mathrm{k,i}}^{\mathrm{a}}(\mathrm{t})\right)\mathbf{w}(t)\right|^2}{N_0\underline{B}_k} \tag{6}$$

is the RIS-dependent normalized downlink channel coefficient, where $\underline{\boldsymbol{h}}_k^a(t) \in \mathbb{C}^{1\times N_i}$, $\underline{\boldsymbol{h}}_{k,i}(t) \in \mathbb{C}^{1\times N_i}$, $\underline{\boldsymbol{G}}_{k,i}^a(t) \in \mathbb{C}^{N_i\times N_a}$, and $\underline{B}_k$ are the downlink counterparts of the parameters in (4). Then, our goal is to optimize the uplink and downlink transmitted powers $\{\overline{p}_k(t)\}_{k=1}^K$ and $\{\underline{p}_k(t)\}_{k=1}^K$, respectively, jointly with the reflectivity parameters $\{\boldsymbol{\nu}_i(t)\}_{i=1}^I$ of the available RISs and the beamformer $\mathbf{w}(t)$ of the AP.

2.2 Evolution of data and computation queues

We assume that each device k generates $A_k(t)$ bits as input of the application to be executed at each time slot t. A queueing system is used to model and control the dynamic data generation, transmission, and processing. In particular, at each time slot t, each user buffers data in a local queue $Q^l(t)$ and send them to the AP at the transmission rate $\overline{R}_k(t)$ (cf. (3)). Thus, the local queue update follows the rule:

$$Q_k^l(t+1) = \max\left(0, Q_k^l(t) - \tau\overline{R}_k(t)\right) + A_k(t). \tag{7}$$

The AP receives data from each device k and sends the data to the ES, which processes J_k bits-for-cycle, where J_k is a parameter that depends on the application offloaded by device k. The ES provides a total CPU frequency $f_c(t)$ at each time slot, and a percentage $f_k(t)$ of it is allocated to process the task offloaded by user k at slot t, such that $\sum_{k=1}^K f_k(t) \leq f_c(t)$. The remote queue at the ES evolves as:

$$Q_k^r(t+1) = \max\left(0, Q_k^r(t) - \tau f_k(t)J_k\right) + \min\left(Q_k^l(t), \tau\overline{R}_k(t)\right), \tag{8}$$

where $\tau f_k(t)J_k$ are bits processed during each slot for UE k. Finally, the AP sends back to each user the bits resulting from the computation. Downlink communications can be incorporated considering an additional queue at the AP. To this aim, we assume that there is a linear dependence among the number of bits in input and those produced in output by the application running at the ES. Let denote by c_k the ratio between output and input bits of the application required by user k. Thus, similarly to the previous models, the processed data are buffered in a queue $Q_k^a(t)$ at the AP and transmitted at a downlink rate $\underline{R}_k(t)$ (cf. (5)), with the update rule:

$$Q_k^a(t+1) = \max\left(0, Q_k^a(t) - \tau \underline{R}_k(t)\right) + c_k \cdot \min\left(Q_k^r(t), \tau f_k(t) J_k\right). \tag{9}$$

In this paper, we perform a joint optimization of RISs phase shifts, radio (uplink and downlink) and computation resources, considering the sum of communication and computation queues, i.e.,

$$Q_k^{\text{tot}}(t) = Q_k^l(t) + Q_k^r(t) + Q_k^a(t), \tag{10}$$

as a metric to quantify the overall delay experienced by data offloaded by each device. As we will show in the sequel, our aim is to keep the average value of $Q_k^{\text{tot}}(t)$ in (10) (which is related to the average service delay through the Little's law [37]) below a given threshold.

2.3 Energy consumption

In this paragraph, we model the overall energy consumption of the system. In particular, we consider the energy spent for computation by the edge server, the energy spent for downlink communications by the AP, the energy spent for uplink transmission by each device and, finally, the energy spent by RISs to shape the wireless environment.

1) ES's energy consumption At the ES, the energy spent for computation (assuming a CMOS-based CPU) is [38]:

$$e_c(t) = \tau \gamma_c \left(f_c(t)\right)^3 + \tau_s \gamma_c f_m^3, \tag{11}$$

where $f_c(t)$ and γ_c are the CPU frequency and the effective switched capacitance of the ES processor, respectively. In (11), we assumed (with the last term) that, during the portion τ_s of the slot, the server performs computations at speed f_m, which represents the minimum CPU frequency necessary to solve the optimization problem we will present in the sequel.

2) AP's energy consumption For the AP energy consumption, we exploit the concept from [13, 39], considering that a large portion of its energy is consumed only for being in *active state* (i.e., to switch on RF chain, power amplifiers, power supply, analog front-end, digital baseband, and digital control). Then, we assume that the AP is able to enter low power sleep operation mode to save energy, a concept known as Discontinuous Transmission (DTX). In particular, let us denote by p_a^{on} the overall power consumption of the AP for being in active state. While in active state, the AP can transmit and/or receive. Instead, in sleep state, the AP can neither transmit nor receive. In active state, for downlink transmissions, the AP provides a maximum total power P_a at each time slot, and a percentage $\underline{p}_k(t)$ of power is allocated for communicating with user k at slot t, such that $\sum_{k=1}^{K} \underline{p}_k(t) \leq P_a$. In [39], four possible Sleep Modes (SM) are defined, with different minimum sleep periods, corresponding to the ODFM symbol, the sub-frame duration, the radio frame duration, and a standby mode. Here, we assume that the kind of sleep mode is selected a priori, while the choice of when being active or sleeping is performed online. To control the active and sleep state of the AP, we introduce the binary variable $I_a(t) \in \{0, 1\}$, which is equal to 1 if and only if the AP is in active state at time slot t. Also, in each time slot, the AP is forced to be active for the first τ_s seconds to perform channel state information (CSI) acquisition and control signaling, which we assume to be performed with the minimum power $\underline{P}_m$ required to achieve the target

estimation and communication performance. The AP energy consumption at time slot t is then given by:

$$e_a(t) = \tau\left(I_a(t)p_a^{\text{on}} + I_a(t)\sum_{k=1}^{K}\underline{p}_k(t) + (1 - I_a(t))p_a^{\text{s}}\right) + \tau_s\left(p_a^{\text{on}} + \underline{P}_m\right), \tag{12}$$

where p_a^{s} represents the (low) power consumed in sleep mode.

3) UE's energy consumption Beyond uplink transmissions, we assume that each UE performs control signaling and channel estimation (during the first τ_s seconds) using the minimum power $\overline{P}_k$ needed to obtain a desired performance. Then, the energy spent by user k at time t is given by:

$$e_k(t) = \tau\,\overline{p}_k(t)I_a(t) + \tau_s\overline{P}_k, \tag{13}$$

for $k = 1, \ldots, K$, where $\overline{p}_k(t)$ affects the uplink data rate as in (3). From (13), if the AP is in sleep mode at time t, user k does not spend energy for uplink transmission.

4) RIS's energy consumption The power consumption of an RIS depends on the type, the resolution, and the number of its individual reflecting elements that effectively perform phase shifting on the impinging signal [28, 40, 41]. In particular, let $p^r(b_i)$ be the power dissipated by each of the N_i phase shifter of RIS i, assuming b_i-bit resolution. Typical power consumption values of each phase shifter are 1.5, 4.5, 6, and 7.8 mW for 3-, 4-, 5-, and 6-bit resolution phase shifting [28]. In every slot, each RIS is forced to have all active elements for the first τ_s seconds to perform CSI acquisition. Of course, if the AP is in sleep mode at time t, also the RIS is switched off. Then, the overall energy spent by RIS i is:

$$e_i^r(t) = I_a(t)\tau p^r(b_i)\sum_{l=1}^{N_i}|v_{i,l}(t)|^2 + \tau_s N_i p^r(b_i), \tag{14}$$

for $i = 1, \ldots, I$, where we exploited the fact that each phase shift coefficient $v_{i,l}$ in (2) has either zero (if the l-th element is off) or unitary (if the l-th element is on) modulus. Thus, from (14), we can control the overall energy spent by the RISs at each time slot, acting on the number of active reflecting elements, and the value of the state variable $I_a(t)$.

In the following section, we will formulate the proposed dynamic strategy for RIS-empowered wireless network edge optimization, aimed at performing energy-efficient dynamic edge computing with guaranteed latency requirements.

3 Problem formulation and methodology

Our goal is to find the optimal scheduling of RISs' parameters (i.e., phase shifts), radio (i.e., powers, rates, AP/RISs/UE duty cycles) and computation (i.e., CPU cycles) resources that minimizes the long-term average of a weighted sum of the energy consumption terms in (11)–(14), under constraints on the maximum average queue length in (10). To this aim, we define the weighted sum energy as follows:

$$e_\sigma^{\text{tot}}(t) = \sigma\sum_{k=1}^{K}e_k(t) + (1-\sigma)\left(e_c(t) + e_a(t) + \sum_{i=1}^{I}e_i^r(t)\right), \tag{15}$$

where $\sigma \in [0,1]$ is a weighting parameter to be chosen. For instance, choosing $\sigma = 1$ leads to a pure *user-centric* strategy; whereas, $\sigma = 0$ induces a pure *network-centric strategy*. An intermediate strategy, which we term as *holistic*, can be obtained with $\sigma = 0.5$. The use of this weighting parameter introduces more degrees of freedom and flexibility in the resource optimization, depending on the needs of operators, users, and service providers. Using (15), the problem can be formulated as:

$$
\begin{aligned}
&\min_{\Psi(t)} \quad \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\{e_{\sigma}^{\text{tot}}(t)\right\} \\
&\text{subject to} \qquad (a) \quad \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\{Q_k^{\text{tot}}(t)\right\} \le Q_k^{\text{avg}}, \quad \forall k; \\
&\qquad\qquad \left.\begin{array}{ll}
(b) & I_a(t) \in \{0,1\} \quad \forall t; \\
(c) & \mathbf{w}(t) \in \mathcal{C} \quad \forall t; \\
(d) & v_{i,l}(t) \in \mathcal{S}_i \qquad \forall i,l,t; \\
(e) & |v_{i,l}(t)|^2 \le I_a(t) \qquad \forall i,l,t; \\
(f) & 0 \le \overline{p}_k(t) \le P_k\, I_a(t), \qquad \forall k,t; \\
(g) & \underline{p}_k(t) \ge 0, \qquad \forall k,t; \\
(h) & \sum_{k=1}^{K} \underline{p}_k(t) \le P_a\, I_a(t), \quad \forall t; \\
(i) & f_k(t) \ge 0, \qquad \forall k,t; \\
(j) & \sum_{k=1}^{K} f_k(t) \le f_c(t), \quad \forall t; \\
(k) & f_c(t) \in \mathcal{F} \quad \forall t;
\end{array}\right\} \mathcal{X}(t)
\end{aligned}
\tag{16}
$$

where $\mathbf{\Psi}(t) = \left[I_a(t), \mathbf{w}(t), \{\mathbf{v}_i(t)\}_{i=1}^{I}, \{\overline{p}_k(t)\}_{k=1}^{K}, \{\underline{p}_k(t)\}_{k=1}^{K}, \{f_k(t)\}_{k=1}^{K}, f_c(t)\right]$, and the expectations are taken with respect to the random channel states and data arrivals, whose statistics are supposed to be unknown. The constraints of (16) have the following meaning: (*a*) the average queue lengths[1] do not exceed a predefined value Q_k^{avg}, for all k; (*b*) the state variable $I_a(t)$ is binary; (*c*) the beamforming vector $\mathbf{w}(t)$ can be selected from a codebook of available combiners/precoders; (*d*) the RIS reflection coefficients take values from the discrete set $\mathcal{S}_i$ in (2); (*e*) RIS modules can be active only if $I_a(t) = 1$; (*f*) the uplink transmission power is greater than zero and upper bounded by $P_k I_a(t)$, for all k; (*g*) the downlink transmission power is greater than zero; (*h*) the sum of all downlink transmitted powers is less than or equal to the maximum power P_a, or 0 whenever the AP is inactive ($I_a(t) = 0$); (*i*) the CPU frequencies assigned to each device are greater than zero, and (*j*) their sum is less than or equal to the ES CPU frequency $f_c(t)$; (*k*) the ES CPU frequency takes values from a discrete set $\mathcal{F}$. Solving (16) is very challenging, because of the lack of knowledge of the statistics of the radio channels and task arrivals, and the inherent non-convexity. A further difficulty is related to the fact that the

[1] More sophisticated constraints can also be imposed on the maximum tolerable delay [11].

RISs are being optimized to handle, simultaneously, multiple data flows. Nevertheless, in the sequel, we will show how these problems can be effectively tackled resorting to stochastic Lyapunov optimization [42], a powerful tool able to first transform long-term constraints into pure stability ones, through the definition of suitable state variables, to finally solve the problem in a per-slot fashion, through the solution of successive deterministic problems in each time slot. This allows us to dramatically simplify the problem, thus deriving low-complexity yet effective solutions in each time slot. As it will be clarified, the per-slot method comes with theoretical guarantees on convergence and asymptotic optimality of the solution, reached through a single tuning parameter used to explore the desired trade-off between energy consumption and E2E delay.

4 RIS-empowered dynamic edge computing based on Lyapunov stochastic optimization

As already mentioned, we now convert the long-term optimization in (16) into a stability problem, hinging on stochastic Lyapunov optimization [42]. The first step is to define suitable state variables, known as *virtual queues*, whose long-term stability guarantees the constraints. More specifically, to deal with the long-term constraints in (a), we introduce K virtual queues that evolve as follows:

$$Z_k(t+1) = \max\left\{0, Z_k(t) + \epsilon_k\left(Q_k^{\text{tot}}(t+1) - Q_k^{\text{avg}}\right)\right\}, \tag{17}$$

$k = 1, \ldots, K$, where $\{\epsilon_k\}_{k=1}^K$ are positive step sizes used to control the convergence speed of the algorithm. A virtual queue is a mathematical model that shows how the system is behaving in terms of constraint violations. Intuitively speaking, if a virtual queue grows too fast, the associated constraints are being violated and the system is not stable. Formally speaking, this translates into the *mean rate stability* of the queues[2], which is equivalent to satisfy the constraints (a) in (16) [42]. To this aim, we first define the Lyapunov function as $\mathcal{L}(t) = \mathcal{L}(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{k=1}^K Z_k^2(t)$, where $\mathbf{\Theta}(t) = \{Z_k(t)\}_{k=1}^K$, and then the *drift-plus-penalty* function given by [42]:

$$\Delta^p(t) = \mathbb{E}\Big\{\mathcal{L}(t+1) - \mathcal{L}(t) + V \cdot e_\sigma^{\text{tot}}(t)\Big|\ \mathbf{\Theta}(t)\Big\}. \tag{18}$$

The drift-plus-penalty function is the conditional expected change of $\mathcal{L}(t)$ over successive slots, with a penalty factor that weights the objective function of (16), with a weighting parameter V. Then, following stochastic optimization arguments as in [42], we proceed by minimizing an upper-bound of the drift-plus-penalty function in (18) in a stochastic fashion. After some algebraic manipulations (similar to the ones used in [13]), we obtain the following per-slot problem at each time t:

[2] A queue $X(t)$ is mean-rate stable if $\lim_{T\to\infty} \mathbb{E}\{X_T\}/T = 0$.

$$\min_{\boldsymbol{\Psi}(t)\in\widetilde{\mathcal{X}}(t)} \sum_{k=1}^{K}\Big[\big(Q_k^r(t)-Q_k^l(t)-Z_k(t)\big)\tau\overline{R}_k(t) + \big(c_kQ_k^a(t)-Q_k^r(t)-Z_k(t)\big)\tau f_k(t)J_k-\big(Q_k^a(t)+Z_k(t)\big)\tau\underline{R}_k(t)\Big]+V\cdot e_\sigma^{\text{tot}}(t) \tag{19}$$

where $\widetilde{\mathcal{X}}(t)$ is the instantaneous feasible set, as defined in (16), with the following modifications: (i) constraint (*e*) becomes $0\le\overline{p}_k(t)\le\widetilde{P}_k(t)I_a(t)$, where $\widetilde{P}_k(t)=\min(P_k,\overline{P}_k(t))$, with $\overline{P}_k(t)$ denoting the minimum power needed to empty the local queue $Q_k^l(t)$ at time t; (ii) constraint (*f*) becomes $0\le\underline{p}_k(t)\le\underline{P}_k(t)I_a(t)$, where $\underline{P}_k(t)$ is the minimum power needed to empty the downlink queue $Q_k^a(t)$ at time t; (iii) constraint (*h*) becomes $0\le f_k(t)\le Q_k^r(t)/\tau J_k$. What is worth to emphasize about (19) is that, as opposed to (16), it does not involve any expectation and it is only based on the current values of the channel and task parameters, as well as on the (virtual and real) queues' states. Because of the structure of set $\widetilde{\mathcal{X}}(t)$, (19) is a mixed-integer nonlinear optimization problem, which might be very complicated to solve. Nevertheless, in the sequel, we will show how (19) can be split into sub-problems that admit low-complexity solution procedures for the optimal RIS parameters (i.e., the phase shifts of its elements), the AP's beamformer, the uplink and downlink radio resources (i.e., powers, sleep mode and duty cycle), and the computation resources at the ES (i.e., CPU clock frequencies).

4.1 Dynamic radio resource allocation and RISs optimization

The radio resource allocation problem aims at optimizing the AP duty cycle variable $I_a(t)$, the beamforming vector $\mathbf{w}(t)$, the uplink and downlink transmission powers $\{\overline{p}_k(t)\}_{k=1}^K$, $\{\underline{p}_k(t)\}_{k=1}^K$, respectively, and the RIS reflectivity parameters $\{\boldsymbol{\nu}_i(t)\}_{i=1}^I$. From (3) and (5), it is clear that the presence of RISs couples uplink and downlink resource allocation, since transmission rates are affected by RISs in both directions. From (19), (15), (3), (5), and (16), the radio resource allocation problem reads as:

$$\begin{aligned}\min_{\boldsymbol{\Gamma}}(t)\quad & -\sum_{k=1}^{K}\overline{U}_k(t)\log_2\left(1+\overline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))\overline{p}_k(t)\right)\\ & -\sum_{k=1}^{K}\underline{U}_k(t)\log_2\left(1+\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))\underline{p}_k(t)\right)+V\Bigg[\sum_{k=1}^{K}\left(\sigma\tau\overline{p}_k(t)+(1-\sigma)\tau\underline{p}_k(t)\right)\\ & +(1-\sigma)I_a(t)\tau p_a^{\text{on}}+(1-\sigma)(1-I_a(t))\tau p_a^{\text{s}}+(1-\sigma)\tau\sum_{i=1}^{I}p^r(b_i)\sum_{l=1}^{N_i}|\nu_{i,l}(t)|^2\Bigg]\\ \text{subject to}\quad & I_a(t)\in\{0,1\};\quad \mathbf{w}(t)\in\mathcal{C};\quad 0\le\overline{p}_k(t)\le\widetilde{P}_k(t)I_a(t)\quad\forall k;\\ & \nu_{i,l}(t)\in\mathcal{S}_i,\quad |\nu_{i,l}(t)|^2\le I_a(t)\quad\forall i,l;\\ & 0\le\underline{p}_k(t)\le\underline{P}_k(t),\quad\forall k;\quad\sum_{k=1}^{K}\underline{p}_k(t)\le P_a\,I_a(t);\end{aligned} \tag{20}$$

where $\boldsymbol{\Gamma}(t)=[\boldsymbol{\nu}(t),\{\overline{p}_k(t)\}_{k=1}^K,\{\underline{p}_k(t)\}_{k=1}^K,I_a(t),\mathbf{w}(t)]$, and

$$\overline{U}_k(t)=(Q_k^l(t)-Q_k^r(t)+Z_k(t))\overline{B}_k\tau, \tag{21}$$

$$\underline{U}_k(t) = (Q_k^a(t) + Z_k(t))\underline{B}_k\tau. \tag{22}$$

Problem (20) is non-convex due to the discrete nature of the phase shifts, the beamforming vector, and the active state variable of the AP (i.e., $I_a(t)$); also, the non-convexity comes from the coupling among variables induced by the presence of RISs and the beamforming at the AP. In principle, the global optimum solution of (20) can be achieved through an exhaustive search over all the possible combinations of $\{\boldsymbol{\nu}_i(t)\}_{i=1}^I$, $\mathbf{w}(t)$, and $I_a(t)$, evaluating the optimal uplink and downlink powers, and selecting the set of variables that yields to the lowest value of the objective function in (20). However, the complexity of this approach grows exponentially with the number I of RISs, the maximum number $N = \max_i N_i$ of RIS elements, and the maximum cardinality $S = \max_i |\mathcal{S}_i|$ of the sets $\mathcal{S}_i$ in (2). Since in the dynamic context considered in this paper resource allocation must take place in a very short amount of time, we follow an alternative (albeit simplified) optimization strategy. In particular, let us first notice that we can distinguish between two different cases.

Case 1 : $I_a(t) = 0$. In this case, problem (20) is trivial, since the AP is in sleep state (thus not receiving and transmitting), and so are also the UE and the RISs. Thus, the only feasible solution reads as:

$$\overline{p}_k(t) = 0, \quad \forall k, \quad \underline{p}_k(t) = 0, \quad \forall k, \quad \boldsymbol{\nu}_i(t) = 0, \quad \forall i. \tag{23}$$

In this case, the objective function of (20) boils down to:

$$\Omega(I_a(t) = 0) = V(1-\sigma)\tau p_a^{\mathrm{s}}. \tag{24}$$

The value in (24) must be compared with the value of the objective function obtained in the following second case.

Case 2 : $I_a(t) = 1$. In this case, the AP is available for transmission and/or reception, so that a solution is needed to select the uplink and downlink radio resources and the RIS reflectivity coefficients. In particular, problem (20) translates into the following simplified sub-problem:

$$\begin{aligned}
\min_{\boldsymbol{\Psi}^r(t)} \quad & -\sum_{k=1}^{K} \overline{U}_k(t) \log_2\left(1 + \overline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t))\overline{p}_k(t)\right) \\
& -\sum_{k=1}^{K} \underline{U}_k(t) \log_2\left(1 + \underline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t))\underline{p}_k(t)\right) \\
& + V\left[\sum_{k=1}^{K}\left(\sigma\tau\overline{p}_k(t) + (1-\sigma)\tau\underline{p}_k(t)\right) + (1-\sigma)\tau p_a^{\mathrm{on}}\right. \\
& \left. +(1-\sigma)\tau\sum_{i=1}^{I} p^r(b_i)\sum_{l=1}^{N_i}|\nu_{i,l}(t)|^2\right] \\
& \text{subject to} \quad 0 \le \overline{p}_k(t) \le \widetilde{P}_k(t) \ \ \forall k; \quad \nu_{i,l}(t) \in \mathcal{S}_i \ \ \forall i,l; \quad \mathbf{w}(t) \in \mathcal{C}; \\
& \qquad 0 \le \underline{p}_k(t) \le \underline{P}_k(t), \ \ \forall k; \quad \sum_{k=1}^{K}\underline{p}_k(t) \le P_a.
\end{aligned} \tag{25}$$

To solve (25), we propose a greedy method that first optimizes (25) with respect to the RIS reflectivity parameters $\{\boldsymbol{\nu}_i(t)\}_{i=1}^{I}$ and the AP's beamforming vector $\mathbf{w}(t)$, and then it selects the uplink and downlink powers. Indeed, given a fixed RISs configuration and AP's beamformer (i.e., for a given value of $\boldsymbol{\nu}(t)$ and $\mathbf{w}(t)$), (25) becomes strictly convex and decouples over uplink and downlink, admitting a simple closed form solution for $\{\overline{p}_k(t)\}_{k=1}^{K}$, and a water-filling like expression for $\{\underline{p}_k(t)\}_{k=1}^{K}$. The details of the three optimization steps (i.e., RISs/AP's beamforming, uplink, and downlink) are given next.

4.1.1 RISs and AP's beamforming optimization

To optimize (20) with respect to the RISs configuration and the AP's beamforming, we notice that, for any value of $\overline{p}_k(t)$, if $\overline{U}_k(t) > 0$, the k-th component of the first objective term in (25) is minimized by increasing the normalized channel coefficients $\overline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t))$. A similar argument applies to the k-th component of the second objective term in (25), which is minimized by increasing the normalized channel coefficient $\underline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t))$. Thus, letting $\mathcal{U}(t) = \{k \mid \overline{U}_k(t) > 0\}$, we exploit the following surrogate optimization function:

$$\begin{aligned}\Delta^R(\boldsymbol{\nu}(t), \mathbf{w}(t)) \; = \; & - \sum_{k \in \mathcal{U}(t)} \overline{U}_k(t) \overline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t)) - \sum_{k=1}^{K} \underline{U}_k(t) \underline{\alpha}_k(\boldsymbol{\nu}(t), \mathbf{w}(t)) \\ & + V(1-\sigma)\tau \sum_{i=1}^{I} p^R(b_i) \sum_{l=1}^{N_i} |\nu_{i,l}(t)|^2,\end{aligned} \tag{26}$$

which represents a linear combination of the RIS energy term in (25), weighted by the Lyapunov parameter V, and (negative) RIS-dependent uplink and downlink channel coefficients in (4) and (6), weighted by the terms $\overline{U}_k(t)$ and $\underline{U}_k(t)$ in (21)-(22), which depend on the communication, computing, and virtual queues' states.

Intuitively, minimizing (26), the RISs will be optimized to favor uplink and/or downlink communications (depending on the status of the cumulative parameters $\overline{U}_k(t)$ and $\underline{U}_k(t)$ for each user k), with a penalty on the energy spent for such improvement in communication performance. This is equivalent to a dynamic scheduling of the RIS resources to serve the users over uplink and/or downlink communications, depending on the status of the queues (i.e., $\overline{U}_k(t)$ and $\underline{U}_k(t)$) that quantify the system congestion. In other words, time plays the role of a further degree of freedom for the scheduling of the RISs, which are dynamically assigned by the proposed Lyapunov stochastic optimization procedure to serve uplink or downlink communications of different users. To the best of our knowledge, this queue-based dynamic control of RIS reconfiguration has never been proposed in the literature. Finally, increasing the value of V, the minimization of (26) leads to more sparse solutions for the vector $\boldsymbol{\nu}(t)$, since it might be unnecessary to switch on all the reflecting elements to satisfy the average latency constraint in (16).

Algorithm 1: Greedy optimization of RISs and AP's beamforming

Input: V, $\{p^r(b_i)\}_{i=1}^I$, $\{\overline{U}_k(t)\}_{k\in\mathcal{U}(t)}$, $\{\underline{U}_k(t)\}_{k=1}^K$, $\{\overline{\boldsymbol{h}}_k^a(t)\}_{k=1}^K$, $\{\overline{\boldsymbol{h}}_k(t)\}_{k=1}^K$, $\{\overline{\boldsymbol{G}}_k^a(t)\}_{k=1}^K$, $\{\underline{\boldsymbol{h}}_k^a(t)\}_{k=1}^K$, $\{\underline{\boldsymbol{h}}_k(t)\}_{k=1}^K$, $\{\underline{\boldsymbol{G}}_k^a(t)\}_{k=1}^K$.

for $\mathbf{w}(t)\in\mathcal{C}$ **do**

  Set $\bar{\boldsymbol{v}}_i=\mathbf{0}\ \forall i$

  for $i=1:I$ **do**

    for $l=1:N_i$ **do**

      $\bar{v}_{i,l}=\arg\min_{v_{i,l}\in\mathcal{S}_i}\ \Delta^R(v_{i,l};\bar{\boldsymbol{v}}_{i,-l},\bar{\boldsymbol{v}}_{-i},\mathbf{w}(t))$

    Set $\boldsymbol{v}_i(\mathbf{w}(t))=[\bar{v}_{i,1},\cdots,\bar{v}_{i,N_i}]^T$

  Set $\boldsymbol{v}(\mathbf{w}(t))=\{\boldsymbol{v}_i(\mathbf{w}(t)\}_{i=1}^N$

$\bar{\mathbf{w}}(t)=\arg\min_{\mathbf{w}(t)\in\mathcal{C}}\ \Delta^R(\boldsymbol{v}(\mathbf{w}(t)),\mathbf{w}(t))$

Output: $\bar{\mathbf{w}}(t)$ and $\bar{\boldsymbol{v}}(t)=\boldsymbol{v}(\bar{\mathbf{w}}(t))$

The steps of the proposed greedy method are illustrated in Algorithm 1, which proceeds according to the following rationale. Fixing an AP's beamforming vector $\mathbf{w}(t)\in\mathcal{C}$, the method greedily optimizes the reflectivity vector $\bar{\boldsymbol{v}}_i$ (initialized at zero) of each RIS i, iteratively selecting the coefficient $v_{i,l}\in\mathcal{S}_i$ that minimizes (26), having fixed all the other parameters of RIS i (i.e., $\bar{\boldsymbol{v}}_{i,-l}$) and of the other RISs (i.e., $\bar{\boldsymbol{v}}_{-i}$). This procedure is repeated for all possible beamforming vector $\mathbf{w}(t)\in\mathcal{C}$, in order to find the pair $(\boldsymbol{v}(t),\mathbf{w}(t))$ that greedily minimizes the objective in (26). For each $\mathbf{w}(t)$, this approach requires $O(S\overline{N})$ evaluations of (26), with $\overline{N}=\sum_{i=1}^I N_i$, and leads to a non-increasing behavior of (26) as more RIS reflecting elements are added and optimized. Interestingly, the function in (26) is greedily optimized filling the vectors $\{\boldsymbol{v}_i(t)\}_{i=1}^N$ one element per time, starting from the zero vector (cf. Algorithm 1). Thus, in the first stages of Algorithm 1, the vector $\boldsymbol{v}(t)$ is composed of almost all zeros (i.e., it is highly sparse), and the computation of (26) is very light (cf. (4) and (6)). This operation is repeated for all $\mathbf{w}(t)\in\mathcal{C}$, thus requiring $O(S\overline{N}|\mathcal{C}|)$ evaluations of the objective function in (26).

Block optimization of RISs Even if the complexity of the greedy procedure in Algorithm 1 is sufficiently low, in practical scenarios one might still desire an even faster procedure. To this aim, we might divide the N_i modules of RIS i in N_b blocks, where the elements of each block are phase-shifted in the same way. Then, proceeding as in Algorithm 1, each block of RIS i is greedily optimized selecting the phase shift coefficient (equal for each element of the block) that leads to the largest decrease of the surrogate objective in (26). Assuming for simplicity that the number of blocks is the same for all RISs, the complexity of Algorithm 1 is reduced of a factor $\overline{N}/IN_b$, which is paid in terms of an overall reduction of performance. This complexity-performance trade-off will be numerically assessed in Sect. 4.

4.1.2 Uplink radio resource allocation

Once the RIS configuration $\boldsymbol{v}(t)$ and the AP's beamformer $\mathbf{w}(t)$ have been fixed through Algorithm 1, from (25), the uplink radio resource allocation decouples from downlink, and reads as:

$$\min_{\{\overline{p}_k(t)\}_{k=1}^K} -\sum_{k=1}^K \overline{U}_k(t)\log_2\left(1+\overline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))\overline{p}_k(t)\right)+V\sigma\tau\sum_{k=1}^K\overline{p}_k(t)$$
$$\text{subject to} \qquad 0\le\overline{p}_k(t)\le\widetilde{P}_k(t), \quad \forall k. \tag{27}$$

Problem (27) is convex, with an additive strictly convex objective that decouples over the users. Now, imposing the Karush–Kuhn–Tucker (KKT) conditions of (27), it is easy to see that the problem admits a closed form solution for the optimal $\{\overline{p}_k(t)\}_{k=1}^K$. In particular, the set $\mathcal{U}(t)=\{k\,|\,\overline{U}_k(t)>0\}$ previously used in (26) takes the role of the set of transmitting users. Indeed, from a rapid inspection of (27), it is clear that user k does not transmit (i.e., $\overline{p}_k(t)=0$) if $\overline{U}_k(t)<0$ (since both terms of the objective function in (27) are monotone non-decreasing functions of $\overline{p}_k(t)$). Thus, we get a simple closed form solution for the optimal uplink powers:

$$\overline{p}_k(t)=\begin{cases}\left[\dfrac{\overline{U}_k(t)}{V\tau\log 2}-\dfrac{1}{\overline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))}\right]_0^{\widetilde{P}_k(t)}, & \text{if } k\in\mathcal{U}_t;\\ 0, & \text{if } k\notin\mathcal{U}_t.\end{cases} \tag{28}$$

As expected, for all k, the transmission powers at time t in (28) are affected by the RIS-dependent uplink channel coefficient $\overline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))$, and the status of the communication, computation, and virtual queues embedded into $\overline{U}_k(t)$ (cf. (21)).

4.1.3 Downlink radio resource allocation

Once the RISs configuration $\boldsymbol{\nu}(t)$ and the AP's beamformer have been fixed, the radio resource allocation problem optimizes the downlink transmission powers $\{\underline{p}_k(t)\}_{k=1}^K$. From (25), we obtain:

$$\min_{\{\underline{p}_k(t)\}_{k=1}^K} -\sum_{k=1}^K \underline{U}_k(t)\log_2\left(1+\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))\underline{p}_k(t)\right)+V(1-\sigma)\tau\left(\sum_{k=1}^K\underline{p}_k(t)+p_a^{\text{on}}\right)$$
$$\text{subject to} \quad 0\le\underline{p}_k(t)\le\underline{P}_k(t), \quad \forall k; \quad \sum_{k=1}^K\underline{p}_k(t)\le P_a. \tag{29}$$

Problem (29) is convex, and its solution can be found very efficiently imposing the KKT conditions. In particular, the Lagrangian associated with (29) reads as:

$$L=-\sum_{k=1}^K \underline{U}_k(t)\log_2\left(1+\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))\underline{p}_k(t)\right)+V(1-\sigma)\tau\left(\sum_{k=1}^K\underline{p}_k(t)+p_a^{\text{on}}\right)$$
$$-\sum_{k=1}^K\beta_k\underline{p}_k(t)+\sum\nolimits_{k=1}^K\gamma_k(\underline{p}_k(t)-\underline{P}_k(t))+\nu\left(\sum\nolimits_{k=1}^K\underline{p}_k(t)-P_a\right). \tag{30}$$

Then, the KKT conditions are given by:

$$
\begin{aligned}
&i)\ \frac{\partial L}{\partial \underline{p}_k} = -\frac{\underline{U}_k(t)\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))}{\log(2)\left(1+\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))\underline{p}_k(t)\right)} + V(1-\sigma)\tau - \beta_k + \gamma_k + \nu = 0, \quad \forall k;\\
&ii)\ \beta_k \geq 0; \quad \underline{p}_k(t) \geq 0; \quad \beta_k \underline{p}_k(t) = 0, \quad \forall k;\\
&iii)\ \gamma_k \geq 0; \quad \underline{p}_k(t) \leq \underline{P}_k(t); \quad \gamma_k\left(\underline{p}_k(t) - \underline{P}_k(t)\right) = 0, \ \forall k;\\
&iv)\ \nu \geq 0; \quad \sum_{k=1}^{K} \underline{p}_k(t) \leq P_a; \quad \nu\left(\sum_{k=1}^{K} \underline{p}_k(t) - P_a\right) = 0.
\end{aligned}
\tag{31}
$$

Now, let us consider two cases. First of all, if we assume that $\sum_{k=1}^{K} \underline{p}_k(t) < P_a$, we have $\nu = 0$ due to condition *iv*) in (31). Then, from condition *i*), the optimal solution is:

$$
\underline{p}_k(t) = \left[\frac{\underline{U}_k(t)}{V(1-\sigma)\log 2} - \frac{1}{\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))}\right]_0^{\underline{P}_k(t)} \quad \forall k. \tag{32}
$$

This means that, evaluating (32) for all k, if $\sum_{k=1}^{K} \underline{p}_k(t) \leq P_a$, then (32) is also the global optimal solution of (29), since it satisfies all the KKT conditions. In the second case, given (32), if $\sum_{k=1}^{K} \underline{p}_k(t) > P_a$, we must have $\nu > 0$, and the optimal solution of (29) is found by imposing $\sum_{k=1}^{K} \underline{p}_k(t) = P_a$ due to condition *iv*) in (31). In this case, from condition *i*) in (31), the solution of (29) admits a water-filling like structure [43] (whose practical implementation requires at most K iterations). More specifically, the optimal powers read as:

$$
\underline{p}_k(t) = \left[\frac{\underline{U}_k(t)}{[V(1-\sigma)+\nu]\log 2} - \frac{1}{\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))}\right]_0^{\underline{P}_k(t)} \quad \forall k, \tag{33}
$$

where ν is the Lagrange multiplier chosen to satisfy the power budget constraint with equality, i.e., $\sum_{k=1}^{K} \underline{p}_k(t) = P_a$. The overall procedure is summarized in Algorithm 2 and is very efficient. Indeed, in the case the closed-form solution in (32) is such that $\sum_{k=1}^{K} \underline{p}_k(t) \leq P_a$, the procedure stops and the water-filling solution in (33) is not needed.

Algorithm 2: Downlink Radio Resource Allocation

Input: V, $\{\underline{\alpha}_k(\boldsymbol{\nu}(t),\mathbf{w}(t))\}_{k=1}^{K}$, $\{\underline{U}_k(t)\}_{k=1}^{K}$.

Let $\{\underline{p}_k^c(t)\}_{k=1}^{K}$ be the candidate powers obtained as (32).

if $\sum_{k=1}^{K} \underline{p}_k^c(t) \leq p_a^{\max}$ **then**

  Set $\underline{p}_k(t) = p_k^c(t)$ for all k

else

  Compute the optimal value of $\underline{p}_k(t)$ as in (33)

Output: $\{\underline{p}_k(t)\}_{k=1}^{K}$

Overall procedure for radio resource allocation Using Algorithm 1, (28), and Algorithm 2, we have the proposed solution to problem (25), i.e., the solution of problem (20) when the AP is active, i.e., $I_a(t) = 1$. Now, to decide the AP state variable $I_a(t)$, we need to compare the value of the objective function of (20) in the active case with the one achieved in the sleep state, i.e., (24). Then, denoting by $\boldsymbol{\nu}^{\text{on}}(t)$, $\mathbf{w}^{\text{on}}(t)$,

$\{\underline{p}_k^{\text{on}}\}_{k=1}^K$, and $\{\overline{p}_k^{\text{on}}\}_{k=1}^K$ the solution obtained with $I_a(t)=1$ (through Algorithm 1, (28), and Algorithm 2), the objective (20) reads as:

$$\begin{aligned}\Omega(I_a(t)=1) = &-\sum_{k=1}^{K}\overline{U}_k(t)\log_2\left(1+\overline{\alpha}_k(\boldsymbol{\nu}^{\text{on}}(t),\mathbf{w}^{\text{on}}(t))\overline{p}_k^{\text{on}}(t)\right)\\ &-\sum_{k=1}^{K}\underline{U}_k(t)\log_2\left(1+\underline{\alpha}_k(\boldsymbol{\nu}^{\text{on}}(t),\mathbf{w}^{\text{on}}(t))\underline{p}_k^{\text{on}}(t)\right)+V\tau\left[\sum_{k=1}^{K}\left(\sigma\overline{p}_k^{\text{on}}(t)+(1-\sigma)\underline{p}_k^{\text{on}}(t)\right)\right.\\ &\left.+(1-\sigma)\left(p_a^{\text{on}}+\sum_{i=1}^{I}p^R(b_i)\sum_{l=1}^{N_i}|v_{i,l}^{\text{on}}(t)|^2\right)\right].\end{aligned} \tag{34}$$

Then, the final solution of (20) is found by comparing (24) and (34). Indeed, if $O(I_a(t)=0)\le O(I_a(t)=1)$, the solution is given by (23). Otherwise, the solution is given by Algorithm 1, (28), and Algorithm 2. The overall procedure for dynamic radio resource allocation is described in Algorithm 3.

Algorithm 3: Dynamic Radio Resource Allocation

Input: All the inputs of Algorithms 1 and 2.

Compute the objective Ω_0 as in (24).

Evaluate:
- $\boldsymbol{v}_i^{\text{on}}(t),\forall i$, and $\mathbf{w}^{\text{on}}(t)$ with Algorithm 1;
- $\overline{p}_k^{\text{on}}(t),\forall k$, as in (28);
- $\underline{p}_k^{\text{on}}(t),\forall k$, with Algorithm 2.

Compute the objective Ω_1 as in (34).

if $\Omega_0\le\Omega_1$ **then**

$I_a(t)=0;\quad \overline{p}_k(t)=\underline{p}_k(t)=0,\forall k;\quad \boldsymbol{v}_i=0,\forall i;$

else

$I_a(t)=1;\quad \overline{p}_k(t)=\overline{p}_k^{\text{on}}(t),\quad \underline{p}_k(t)=\underline{p}_k^{\text{on}}(t),\forall k;\quad \boldsymbol{v}_i(t)=\boldsymbol{v}_i^{\text{on}}(t),\forall i;\quad \mathbf{w}(t)=\mathbf{w}^{\text{on}}(t);$

Output: $I_a(t)$, $\mathbf{w}(t)$, $\{\overline{p}_k(t)\}_{k=1}^K$, $\{\underline{p}_k(t)\}_{k=1}^K$, $\{\boldsymbol{v}_i(t)\}_{i=1}^I$

4.2 Dynamic allocation of computing resources

The computing resource allocation problem optimizes the CPU frequencies $\{f_k(t)\}_{k=1}^K$ assigned by the server to the devices, and the overall ES frequency $f_c(t)$. From (19), letting $Y_k(t)=(-c_kQ_k^a(t)+Q_k^r(t)+Z_k(t))J_k\tau$, we obtain

$$\begin{aligned}\min_{\{f_k(t)\}_{k=1}^K,f_c(t)}\quad &-\sum_{k=1}^{K}Y_k(t)f_k(t)+V(1-\sigma)\tau\gamma_s(f_c(t))^3\\ \text{subject to}\quad &0\le f_k(t)\le\frac{Q_k^r(t)}{\tau J_k},\quad\forall k;\\ &\sum_{k=1}^{K}f_k(t)\le f_c(t);\qquad f_c(t)\in\mathcal{F}.\end{aligned} \tag{35}$$

The CPU frequency $f_c(t)$ in (35) is assumed to belong to a fixed discrete set $\mathcal{F}$. Thus, for a given $f_c(t)\in\mathcal{F}$, problem (35) is linear in $\{f_k(t)\}_{k=1}^K$, and can be solved using the

simple procedure in Algorithm 4. Intuitively, Algorithm 4 assigns the largest portions of $f_c(t)$ to the devices with largest values of $Y_k(t)$, and requires at most K steps. Also, letting $\mathcal{C}(t) = \{k \mid Y_k(t) > 0\}$, it is clear that the ES assigns a nonzero CPU frequency only to the devices belonging to $\mathcal{C}(t)$. Finally, denoting by $\{f_k(f_c(t))\}_{k=1}^K$ the optimal frequencies assigned at the users for a given $f_c(t) \in \mathcal{F}$ (using Algorithm 4), the optimal ES frequency $f_c(t)$ is given by:

$$f_c(t) = \arg\min_{f_c \in \mathcal{F}} \; - \sum_{k \in \mathcal{C}(t)} Y_k(t) f_k(f_c) + V(1-\sigma)\tau\gamma_s(f_c)^3. \tag{36}$$

The variables $f_c(t)$ and $\{f_k(f_c(t))\}_{k=1}^K$ represent the global optimal solution of (35) at time t. The worst case number of scalar operations needed by this procedure is $O(K|\mathcal{F}|)$, which is affordable in many practical scenarios.

Algorithm 4: Optimal scheduling of CPU frequencies

Input: $\{Y_k(t)\}_k$, $\mathcal{C}(t)$, $\{Q_k^r(t)\}_k$, $\{J_k\}_k$, f_c, K.

Set $f_{\text{av}} = f_c$, $\{f_k\}_{k=1}^K = 0$, and $\mathcal{C} = \mathcal{C}(t)$

while $f_{\text{av}} > 0$ **do**

- Find $\widetilde{k} = \arg\max_{k \in \mathcal{C}} \; Y_k(t)$
- Set $f_{\widetilde{k}} = \min\left(\frac{Q_{\widetilde{k}}^r(t)}{\tau J_{\widetilde{k}}}, f_{\text{av}}\right)$
- Set $\mathcal{C} = \mathcal{C} - \left\{\widetilde{k}\right\}$; $f_{\text{av}} = f_{\text{av}} - f_{\widetilde{k}}$
- If $\mathcal{C} = \emptyset \rightarrow$ break

Output: $\{f_k\}_{k=1}^K$

4.3 Overall algorithmic solution

The overall procedure for the proposed resource allocation strategy for RIS-empowered dynamic mobile edge computing is summarized in Algorithm 5, which explains how the previously introduced Algorithms 1-4 are intertwined. In particular, the first step of Algorithm 5 aims at allocating radio resources exploiting Algorithm 3, which embeds Algorithms 1 and 2 (cf. Algorithm 3). Then, the second step of Algorithm 5 hinges on Algorithm 4 to allocate the computing resources of the ES. The first two steps of Algorithm 5 (i.e., the resource allocation) are run by the ES, which is the only entity assumed to have the overall knowledge of the system in terms of, e.g., channels, queues, etc. Also, since steps 1 and 2 of Algorithm 5 involve decoupled optimizations over radio and computing variables, respectively, they can also be computed in parallel to make the implementation more efficient.

Clearly, Algorithm 5 builds on the previously derived joint optimization of communication, computation, and RISs parameters in Sects. 3.1 and 3.2. The method is fully dynamic and optimizes variables on-the-fly via closed form expressions or low-complexity procedures (which do not require asymptotic convergence of iterative algorithms), based on instantaneous realizations and observation of all involved random variables (i.e., wireless channels, and data arrivals), as well as that of physical and virtual queue

states. Algorithm 5 is run at the ES, and the optimized variables are then sent to the UE, AP, and RISs, within the portion τ_s of the time slot dedicated to resource optimization and signaling. Distributed implementations to limit the exchange of state information can also be envisioned, but are beyond the scope of this paper.

Algorithm 5: : RIS-empowered dynamic MEC

Set the Lyapunov trade-off parameter V, $Z_k(0)$, ϵ_k, for all k. In each time slot $t \geq 0$, repeat the following steps:

1. Find the RISs phase shifts $\{\boldsymbol{v}_i\}_{i=1}^{I}$ and the radio parameters $\mathbf{w}(t)$, $I_a(t)$, $\{\overline{p}_k(t)\}_{k=1}^{K}$, $\{\underline{p}_k(t)\}_{k=1}^{K}$ using Algorithm 3;
2. Solve the CPU scheduling problem in (36), where the optimal frequencies $\{f_k(t)\}_{k=1}^{K}$ assigned by the edge server are given by Algorithm 4.
3. Perform the mobile edge computing task.
4. Update the physical queues as in (7), (8), (9), and the virtual queues as in (17).

Interestingly, the algorithm comes with theoretical guarantees in terms of stability and performance. In particular, the following proposition holds.

Proposition 1 *Suppose that the channel gains* $\{\overline{\boldsymbol{h}}_k^a(t)\}_k$, $\{\overline{\boldsymbol{h}}_{k,i}(t)\}_{k,i}$, $\{\overline{\boldsymbol{G}}_{k,i}^a(t)\}_{k,i}$, $\{\underline{\boldsymbol{h}}_k^a(t)\}_k$, $\{\underline{\boldsymbol{h}}_{k,i}(t)\}_k$, $\{\underline{\boldsymbol{G}}_{k,i}^a(t)\}_{k,i}$, *and the data arrivals* $\{A_k(t)\}_k$ *are i.i.d over time. Then, let* (16) *be feasible, and* $\mathbb{E}\{\mathcal{L}(\Theta(0))\} < \infty$. *Then, Algorithm* 5 *guarantees that all physical and virtual queues are mean-rate stable and that the network objective* $e_{\text{tot}}^{\sigma}(t)$ *satisfies*:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{e_{\text{tot}}^{\sigma}(t)\} \leq e_{\text{tot}}^{\sigma,opt} + \frac{\zeta + C}{V}, \tag{37}$$

where $e_{\text{tot}}^{\sigma,opt}$ *is the infimum time average energy achievable by any policy that meets the required constraints, and C and* ζ *are finite positive constants.*

1 ***Proof***

The claim follows from the fact that the control policy given by Algorithm 5 is a C-additive approximation [42, p. 59], which admits inexact solutions (with bounded error) of the drift-plus-penalty method in (19) at each time t. In fact, the solution of Algorithm 5 generally differs from the one of (19), because the greedy Algorithm 1 (embedded into Step 1 of Algorithm 5) is not guaranteed to find the optimal RIS configuration for a given set of channel configurations and data arrivals. However, since the objective and the feasible set of (19) are bounded, for any given value of the (real and virtual) queues at time t, the (expected conditional) difference of the objective values achieved by an exhaustive search procedure (striking the optimum) and the proposed approach in Algorithm 1 is always upper-bounded by a finite constant C. This proves that Algorithm 5 is a C-additive approximation of the drift-plus-penalty method in (19). Finally, following the same approach used in [42, p. 61], since all the functions in (19) are bounded over the feasible set for all t, it exists a finite constant $\zeta > 0$ *such that the main claim of the Proposition comes as a consequence of [42, Theorem 4.8].* □

Proposition 1 guarantees that Algorithm 5 provides stability of the system, while asymptotically approaching the optimal solution of (16) as V increases [42, Th. 4.8]. In practical scenarios with finite V values, the higher is V, the more importance is given to the energy consumption, rather than to the virtual queue backlogs, thus pushing the solution closer to optimality, while still guaranteeing the stability of the system. In Sect. 4, we will numerically assess the performance of the proposed resource allocation strategy.

5 Numerical results and discussion

We consider a scenario similar to Fig. 1, with $K = 5$ users wishing to offload their applications to the ES, through a wireless connection with an AP operating at $f_0 = 28$ GHz. The total available bandwidth is $B = 100$ MHz, equally shared among users, and a noise power spectral density $N_0 = -174$ dBm/Hz. At each time slot, the SISO channels $\{\underline{\boldsymbol{h}}_k^a\}_{k=1}^K$, $\{\overline{\boldsymbol{h}}_k^a\}_{k=1}^K$ between the users and the AP, the channels $\{\underline{\boldsymbol{h}}_{k,i}\}_{i,k}$, $\{\overline{\boldsymbol{h}}_{k,i}\}_{i,k}$ between the users and RISs, and the channels $\{\underline{\boldsymbol{G}}_{k,i}^a\}_{i,k}$, $\{\overline{\boldsymbol{G}}_{k,i}^a\}_{i,k}$ between the RISs and the AP are generated through the available tool SimRIS [36]. In particular, denoting by (x, y, z) the 3D coordinates of an element, we set the following positions and parameters:

- 1 AP at (0, 25, 2), with $p_a^{\text{on}} = 2.2$ W, $p_a^{\text{s}} = 278$ mW and $P_a = 24$ dBm, according to a pico-cell case [39].
- 2 RISs with $N_i = 100$ elements at (33, 28, 2) and (33, 18, 2). We assume that each phase can be encoded with b_i bits (cf. (2)), with b_i ranging from 1 to 3 across the different simulations. Therefore, the energy consumption assumed for controlling a single element is set to (cf. (14)) $p^R(b_i = 1) = 0.5$ mW, $p^R(b_i = 2) = 1$ mW, and $p^R(b_i = 3) = 1.5$ mW.
- 5 users at (34, 20, 1), (35, 20, 1), (36, 20, 1), (36, 22, 1), and (35, 22, 1). The maximum transmit power of a generic user k is set to $P_k = 100$ mW.

All channels experience a coherence time equal to the total slot duration $\tau_l = 10$ ms. A portion $\tau_s = 1$ ms is devoted to control signaling and optimization. Thus, the queues are drained for $\tau = 9$ ms, while the arrival rate is computed as $\bar{A}_k = \mathbb{E}\{A_k(t)/\tau_l\}$ and is set to 100 kbps with Poisson distribution, for all users. As depicted in Fig. 1, we assume that an obstacle obscures the direct communication between the users and the AP with 30 dB of additional path loss. From an application point of view, we consider a conversion factor $J_k = 10^{-3}, \forall k$ (cf. (8)), and $c_k = 1, \forall k$ (cf. (9)). An average constraint on the E2E delay equal to 50 ms is imposed (cf. (16)). Also, we assume that the ES frequency f_t^s (cf. (11)) can be selected in the finite set $\mathcal{F} = [0, 0.01, 0.02, \ldots, 1] \times f_{\max}$, with $f_{\max} = 4.5$ GHz, while the effective switched capacitance of the processor is set to $\gamma_s = 10^{-27}\ W \cdot s^3$.

5.1 Energy-delay trade-off

In this section, we illustrate the trade-off between user energy consumption and E2E delay, obtained with our strategy by tuning the trade-off parameter V (cf. (18)). We first present a single user setting, to then extend it to the multi-user scenario already described. For this first simulation, we consider a user-centric strategy, i.e., $\sigma = 1$ (cf. (15)).

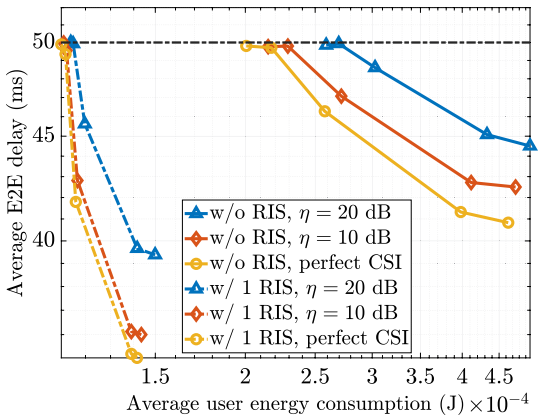


Fig. 2 Average delay versus user energy (single user)

5.1.1 Single user/single RIS

We start from a simple scenario that involves the presence of one user, with an arrival rate $\bar{A}_1 = 500$ kbps, and (possibly) one RIS (i.e., the first ones listed in the simulation setup), and a single antenna AP. In Fig. 2, we show the E2E delay as a function of the user energy consumption, comparing a scenario without RIS with the case where one RIS is exploited, considering also imperfect channel state information (CSI). In particular, we consider both the perfect CSI case, and two cases in which the latter is estimated with an error, with η denoting the estimation signal to noise ratio. The curves in Fig. 2 are obtained by increasing the Lyapunov trade-off parameter V from right to left. As expected, by increasing V, the energy consumption decreases, while the average E2E delay increases up to the desired maximum bound $D_k^{\text{avg}} = 50$ ms, for all the proposed settings. Since this work represents the first contribution on RIS-aided dynamic edge computing, the w/o RIS cases in Fig. 2 represent the current state of the art. Then, from Fig. 2, we can notice how the proposed method exploiting RISs largely outperform the case without RIS in terms of energy-delay tradeoff. Also, the imperfect knowledge of channel states has a small impact on the performance (especially in the RIS aided scenario), thanks to the C-approximation concept introduced in Section 3.1.

To assess the performance boost introduced by multi-antenna communications, in Fig. 3 we report the behavior of the E2E delay as a function of the user energy consumption, comparing a scenario with a multi-antenna AP with the single-antenna case (i.e., SISO). Also, in Fig. 3 we illustrate the behaviors without the presence of RIS (i.e., No RIS), and with random selection of the RIS reflection coefficients (i.e., RIS random), which is known to perform well in rich scattering scenarios [44]. The simulation parameters are: $N_a = 16$ antennas at the AP, additional path loss due to obstacle equal to 20 dB, 1 RIS with 100 elements and 1-bit quantization, and $J_k = 10^{-2}$. The antenna patterns used to build $\mathcal{C}$ are taken from [45], with each element modeled as in [46, Eqn. 2]. A range of $-60°$ to $60°$, with a step of $10°$ is considered to be the codebook of possible beam patterns. As we can notice from Fig. 3, the presence of the RIS optimized according to the proposed strategy (i.e., RIS heuristic) always improves the performance with respect to the no RIS case and to random RIS; instead, the random RIS setting does not bring any benefit in terms of energy-latency trade-off in both SISO or MISO scenarios.

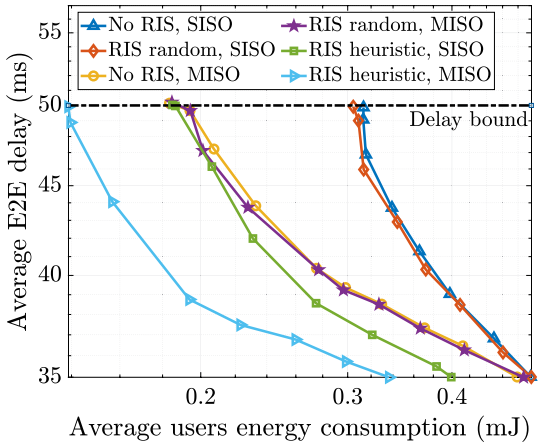


Fig. 3 Average delay vs. user energy for single- and multi-antenna systems

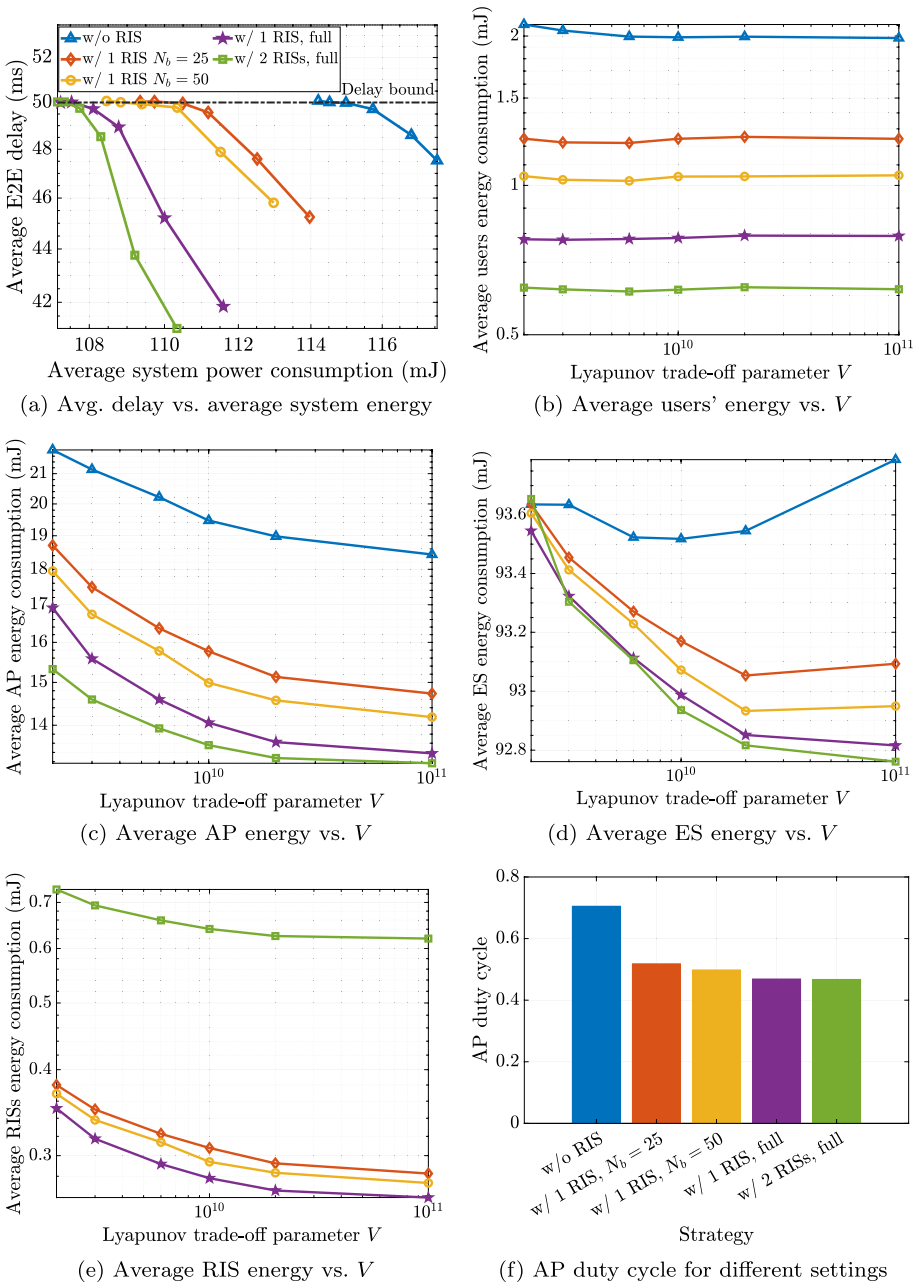
Also, from Fig. 3, it is clear the great advantage introduced by multiple antennas both in the no RIS case and, even more evidently, in the presence of an RIS. Finally, it is quite interesting to notice that, in the proposed setting, the no RIS MISO case has similar performance with respect to the RIS heuristic SISO case, suggesting that an efficient optimization of the RIS can cope with the lack of multiple antennas at the AP; a fact that can greatly reduce deployment and maintenance costs.

5.1.2 Multiple users/multiple RISs

Now, we simulate a more challenging case that encompasses multiple users and possibly multiple RISs, as described in the simulation setup. For this simulation, we consider a holistic strategy that equally weights users and network energy consumption, i.e., $\sigma = 0.5$ (cf. (15)). Thus, in Fig. 4a, we show the E2E delay versus the network energy consumption, considering 5 different conditions:

- A scenario without RISs;
- A scenario with 1 RIS, i.e., the second RIS is switched off. Also, the optimization in Algorithm 1 is performed for each element. We term this strategy as *1 RIS, full*;
- A scenario with 1 RIS, where $N_b = 50$ blocks are defined, i.e., RIS element are optimized in groups of 2. This strategy aims at reducing the complexity of the greedy strategy in Algorithm 1. Thus, given the number of elements N_i, elements are optimized, thorough Algorithm 1, in groups of $\frac{N_i}{N_b}$ elements;
- A scenario with 1 RIS, with 25 optimization blocks, i.e., RIS elements are optimized in groups of 4;
- A scenario with 2 RISs, with full optimization.

The curves in Fig. 4a are obtained by increasing the Lyapunov trade-off parameter V from right to left. By increasing V, each curve reaches a different value of the energy consumption, while converging to the desired delay bound. As expected, all scenarios with RISs outperform the scenario without RIS, with the full optimization (with both 1 and



(a) Avg. delay vs. average system energy

(b) Average users' energy vs. V

(c) Average AP energy vs. V

(d) Average ES energy vs. V

(e) Average RIS energy vs. V

(f) AP duty cycle for different settings

Fig. 4 Energy-delay trade-off in a multi-user settings

2 RISs) achieving the largest gain. The block optimization (with $N_b = 25$ and $N_b = 50$) reduces complexity at the cost of increased energy with respect to the full strategies.

By looking at Fig. 4a, one may conclude that the gain in terms of overall network energy consumption could be negligible. However, this is not true, since we need to analyze the single sources of energy consumption (i.e., users and network elements) separately. In particular, in Fig. 4b, we show the average sum energy consumption of all users as a function of the trade-off parameter V, for the same values used to obtain Fig. 4a. Let us first notice that, while the whole network energy consumption is a monotone

non-increasing function of V, this does not necessarily hold for the single source of energy (users, AP, ES and RISs), due to the fact that we minimize a weighted sum of the energy sources (cf. (15)). But most importantly, from Fig. 4b, we can notice the considerable energy gain in terms of users energy consumption obtained in all the scenarios with RISs, for all values of V, with respect to the scenario without RIS. Also, if we concentrate on the largest value of V, we can compare the strategies for the same maximum average E2E delay (i.e., the bound in Fig. 4a). As a result, from Fig. 4b, the strategy with 2 RISs yields a user energy consumption more than 3 times lower than the value achievable in the non-RIS scenario. In the case of 1 RIS optimized with $N_b = 25$ elements, we obtain around a 30% gain. This reduced gain is the price paid by the complexity reduction with respect to the full optimization. Similar consideration can be made for the AP energy consumption in Fig. 4c, which shows considerable energy gains. This is due to the fact that, since uplink and downlink communications are empowered by the RISs, the users and the AP are able to transmit more data when the AP is active, leaving more time to join the sleep state and save energy (cf. (12)). Thus, the AP duty cycle is reduced by the presence of RISs, as we can see from Fig. 4f, which shows the results of the different strategies for the last value of $V = 10^{11}$.

The effect of RISs is instead less visible on the energy consumption of the server, illustrated in Fig. 4d, which is stable around similar values for all scenarios and for all values of V. Finally, the energy consumption of the RISs is shown in Fig. 4e, where we can notice the increased energy consumption with 2 RISs. Obviously, in the case without RISs, the energy consumption is equal to zero. In summary, the take-home message of Fig. 4 is three-fold:

- Our dynamic strategy is able to reduce the whole energy consumption, with the cost of an increased delay, up to the threshold defined through constraint (a) of (16);
- Empowering MEC with RISs slightly reduces the whole network energy (a non-straightforward fact due to the presence of the RIS energy consumption), while it yields a large gain in terms of users and AP energy consumption.
- The complexity of Algorithm 1 can be reduced by optimizing groups of elements, with the cost of a decreased (yet considerable) gain in terms of energy performance.

5.1.3 Comparison with communication-oriented RIS optimization strategies

An important aspect for the validation of the proposed technique for RIS optimization is the comparison with other methods available in the literature, which mainly optimized RISs to boost communication parameters such as, e.g., rate, signal to noise ratio, or mean-square error of symbol recovery. To this aim, in Fig. 5, we report the behavior of the E2E delay as a function of the user energy consumption, comparing the proposed MEC-oriented RIS-optimization strategy (cf. Algorithm 1) with a purely communication-oriented approach aimed at greedily maximizing the sum of the RIS-aided wireless channel gains. Such competitor can be seen as the customization of the RIS-optimization approach in [32] to our multi-user dynamic MEC scenario with discrete constraints on RIS phase shifts. For this simulation, we consider a SISO system serving two UE, one RIS with 50 elements and 1 bit quantization, an additive

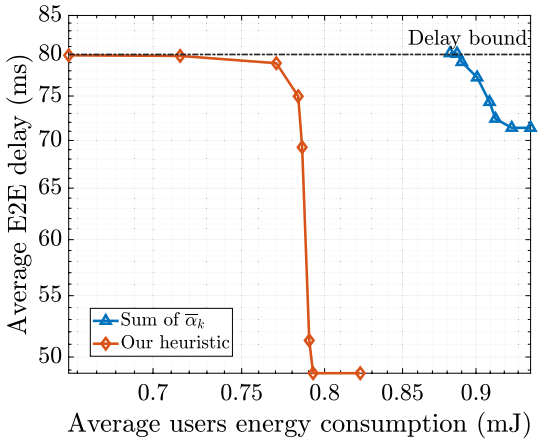


Fig. 5 Average delay vs. users energy for different RIS optimization strategies

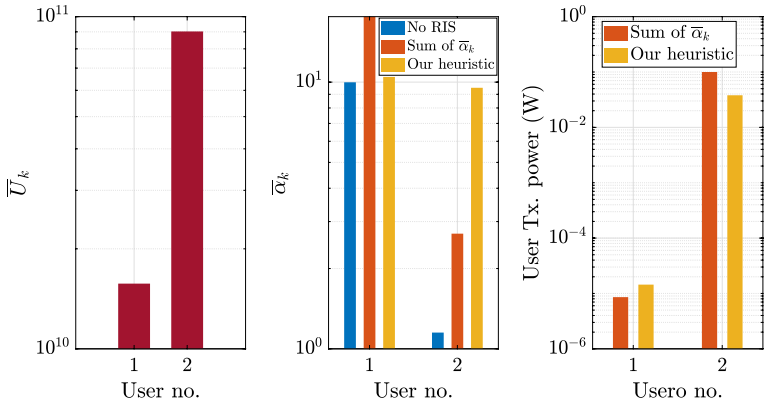


Fig. 6 Instantaneous RIS and radio resource allocation, for different RIS optimization strategies

path loss on the direct path equal to 23 dB, Poisson arrivals with average arrivals 10^2 and 10^5 bits per slot, and $J_k = 10^{-1}$ for both users. As we can notice from Fig. 5, the proposed MEC-oriented strategy largely outperforms the simple communication-oriented approach. This is mainly due to the fact that our strategy incorporates information about the service latency (comprising both communication and computation) thanks to the presence of the queue parameters $\overline{U}_k(t)$ and $\underline{U}_k(t)$ in (26). This positive behavior can also be observed in Fig. 6, which illustrates the power resource allocation over a single time slot of the algorithm, comparing our RIS-optimization method (i.e., our heuristic) with the communication-oriented approach (i.e., sum of channel gains). In particular, on the left side of Fig. 6, we report the queue parameters in (21) for the two UE, which quantify the cumulative congestion of the users from both a communication and computation perspective. As we can see from Fig. 6 (left), user no. 2 has a much larger queue than user no. 1. In this situation, the resource allocation should help user no. 2 to reduce its queue, thus consequently stabilizing the system. Then, in Fig. 6 (center), we report the RIS-aided channel gains obtained in three cases: (i) without the presence of the RIS; (ii) considering the RIS optimized using the communication-oriented strategy; (iii) considering the RIS optimized according to

Table 1 Percentage of saved time with respect to full optimization

b/N_b	1	2	5	10	20	25	50	100
1	0.5%	1%	2%	4%	7%	8%	15%	29%
2	2%	3%	4%	6%	10%	14%	26%	50%
3	1%	2%	5%	10%	22%	23%	54%	100%

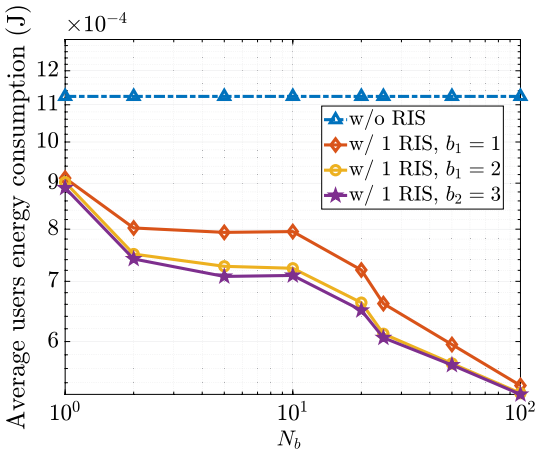


Fig. 7 Average users' energy consumption vs. N_b

our MEC-oriented strategy (i.e., Algorithm 1). As we can notice from Fig. 6 (center), the communication-oriented strategy improves the RIS-aided channel gains of both users in a similar manner with respect to the no RIS case; whereas, our heuristic performs a selective improvement (around 4.5 times) of the RIS-aided channel gain of user no. 2, which is the one having the largest queue. In other words, the RIS has been selectively assigned to the user having the largest queue, thanks to the soft dynamic assignment scheme implemented by Algorithm 1. This gain is then reflected into the power allocation of the users, which is shown in Fig. 6 (right). Indeed, while the power allocation of user 1 is almost the same for both strategies, there is a large reduction of power for the second user. This positive behavior comes from the incorporation of the cumulative queue parameters in (21) and (22) in the objective function (26), which is indeed what leads to the very good performance of our strategy in the context of dynamic MEC empowered by RISs.

5.2 Block optimization of RISs in user-centric scenarios

The results obtained in Fig. 4 motivate us to explore the performance in terms of energy consumption and complexity in the user-centric case (i.e., $\sigma = 1$ in (15)), by varying the number of blocks N_b and the number of bits b_i used to optimize RIS's phases (cf. (2)). To this aim, in Fig. 7, we illustrate the users energy consumption as a function of the number of blocks $N_b = [1, 2, 5, 10, 20, 25, 50, 100]$. Let us recall that $N_b = 100$ corresponds to the full optimization of Fig. 4a, while $N_b = 1$ is the lowest complexity strategy, since it excites all RIS elements with the same phase. For this simulation, we consider only one RIS, and we compare the results with the non-RIS scenario, which is depicted with

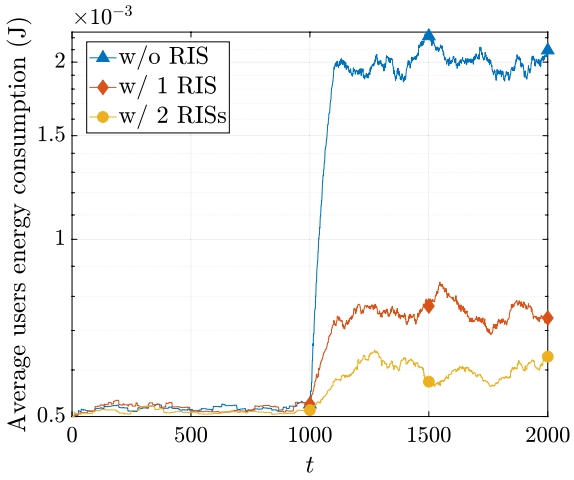
a horizontal line. From Fig. 7, we can notice how, using the RIS is always beneficial, even in the case with $N_b = 1$, although with a slight gain with respect to the non-RIS scenario. As expected, by increasing the number N_b of blocks, the energy consumption decreases thanks to the larger degrees of freedom in optimizing the RIS elements. Also, increasing the number of bits yields a further reduction in the energy consumption, which is more or less appreciable depending on N_b. Finally, from a complexity point of view, we show in Table 1 the percentage of saved time in running a single instance of Algorithm 1, with respect to the highest complexity strategy ($b_1 = 3$, $N_b = 100$). From Table 1, decreasing b_i as well as N_b, we can achieve a considerable gain in terms of computation time needed to find the solution, paid with an increased energy consumption. This quantifies the inherent energy-complexity trade-off introduced by the RIS block optimization.

5.3 Adaptation in non-stationary scenarios

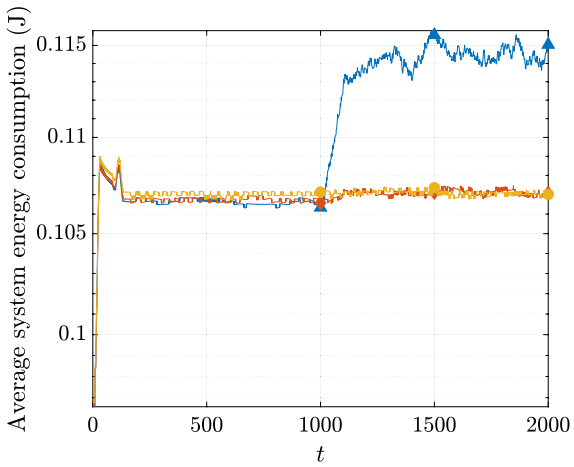
As a final result, we illustrate how the proposed method behaves in a non-stationary scenario with dynamic channel blocking. We assume that, at the beginning of the optimization, no obstacle obscures the direct path between AP and users. Then, at slot number 1000, an obstacle with 30 dB attenuation is interposed in the direct path. For this simulation, we consider again a holistic strategy ($\sigma = 0.5$), and we compare the results without RIS, with 1 RIS and with 2 RISs, with full RIS optimization ($N_b = 100$). Then, in Fig. 8a–c, we illustrate the moving average of the users energy consumption, system energy consumption, and average E2E delay, obtained by averaging these quantities over the last 100 slots. From Fig. a, b, we can notice how, at the beginning, all scenarios converge to a similar user and system energy consumption, due to the fact that the direct path is in good conditions and the RIS does not yield considerable gains. However, when the blockage occurs, the case without RIS is heavily affected from a user energy consumption perspective. This is due to the fact that the direct path is strongly attenuated, which requires higher user transmit power and more AP activity to cope with the arrival rate and stabilize the system. On the contrary, the presence of an RIS determines only a mild effect of a blocking event on the performance. Indeed, as we can see from Fig. 8a, b, with one RIS, the energy consumption is affected due to the blocked direct path, but it is able to converge (in a few time-slots) to a new value much lower than the non-RIS case thanks to the alternative path and the inherent gain of the RIS channel. With two RISs, this gain is even more visible. Finally, from Fig. 8c, we can notice how the delay stabilizes, in both cases, around the threshold, albeit a slight violation caused by the fact that the average is performed over a small window of 100 slots.

6 Conclusions

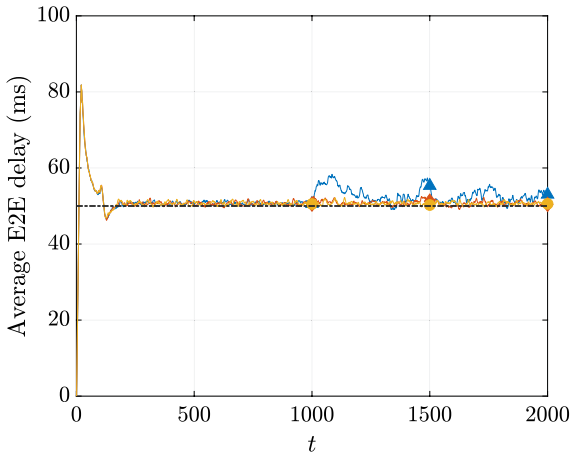
In this paper, we have proposed a novel algorithm for energy-efficient low-latency dynamic edge computing, empowered with reconfigurable intelligent surfaces. The method hinges on stochastic optimization tools, learning dynamically and jointly the phases of RISs elements, the beamforming at the access point, the radio parameter of users and of the access point (i.e., powers and active states), and the CPU frequencies of the edge server. Even in the complex dynamic MEC scenario considered in the paper, the proposed approach requires only low-complexity procedures at each time slot and enables online adaptation of the RISs configuration to dynamically shape the wireless



(a) Average users energy vs. t



(b) Average system energy vs. t



(c) Average delay vs. t

Fig. 8 Temporal behaviors of energy and delay in a non-stationary scenario

propagation channel. Being fully adaptive, the method does not need any a priori knowledge of channel and data arrival statistics. Numerical results assess the performance of the proposed strategy, illustrating the potential gain and adaptation capabilities achievable endowing MEC systems with multiple reconfigurable intelligent surfaces.

Abbreviations

MEC	Mobile edge computing
RIS	Reconfigurable intelligent surface
CPU	Central processing unit
E2E	End-to-end
UE	User equipment
ES	Edge server
AP	Access point
CSI	Channel state information

Acknowledgements
Not applicable.

Author contributions
PDL and MM have contributed to the problem formulation, the algorithmic development, and the practical implementation. ECS and SB have contributed to the problem formulation and the editing of the paper. All authors read and approved the final manuscript.

Authors' Information
Di Lorenzo and Barbarossa are with Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), research unit of Sapienza University of Rome, Italy. Merluzzi and Calvanese Strinati are with CEA Leti, Univ. of Grenoble Alpes.

Funding
The work of Di Lorenzo, Calvanese Strinati, and Barbarossa was supported by the European Union H2020 RISE-6G Project No. 101017011. Barbarossa's work was also supported by MIUR under the PRIN Liquid-Edge contract.

Availability of data and materials
All results are included in this published article. The codes used for generating the results are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 26 May 2022 Accepted: 3 December 2022
Published online: 13 December 2022

References

1. S. Ahmadi, *5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards* (Elsevier Science, Amsterdam, 2019)
2. E. Calvanese Strinati et al., 6G: the next frontier: from holographic messaging to artificial intelligence using subterahertz and visible light communication. IEEE Veh. Technol. Mag. **14**(3), 42–50 (2019)
3. S. Barbarossa, S. Sardellitti, E. Ceci, M. Merluzzi, The edge cloud: a holistic view of communication, computation, and caching. In: Cooperative and Graph Signal Processing (Academic Press, Cambridge, 2018), pp. 419–444
4. A. Ndikumana, N.H. Tran, T.M. Ho, Z. Han, W. Saad, D. Niyato, C.S. Hong, Joint communication, computation, caching, and control in big data multi-access edge computing. IEEE Trans. Mob. Comput. (2019). https://doi.org/10.1109/TMC.2019.2908403
5. M. Merluzzi, P. Di Lorenzo, S. Barbarossa, Wireless edge machine learning: resource allocation and trade-offs. IEEE Access **9**, 45377–45398 (2021)
6. S. Barbarossa, S. Sardellitti, P. Di Lorenzo, Communicating while computing: distributed mobile cloud computing over 5G heterogeneous networks. IEEE Signal Proc. Mag. **31**(6), 45–55 (2014)
7. C. You, K. Huang, H. Chae, B.-H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Trans. Wirel. Commun. **16**(3), 1397–1411 (2016)
8. Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J. Sel. Areas Commun. **34**(12), 3590–3605 (2016)
9. Y. Mao, J. Zhang, S.H. Song, K.B. Letaief, Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. IEEE Trans. Wirel. Commun. **16**(9), 5994–6009 (2017)
10. C. Liu, M. Bennis, M. Debbah, H.V. Poor, Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. IEEE Trans. Commun. **67**(6), 4132–4150 (2019)

11. M. Merluzzi, P. Di Lorenzo, S. Barbarossa, V. Frascolla, Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications. IEEE Trans. Signal Inf. Process. Over Netw. (2020). https://doi.org/10.1109/TSIPN.2020.2981266
12. D. Han, W. Chen, Y. Fang, Joint channel and queue aware scheduling for latency sensitive mobile edge computing with power constraints. IEEE Trans. Wirel. Commun. **19**(6), 3938–3951 (2020). https://doi.org/10.1109/TWC.2020.2979136
13. M. Merluzzi, N. di Pietro, P. Di Lorenzo, E.C. Strinati, S. Barbarossa, Discontinuous computation offloading for energy-efficient mobile edge computing. IEEE Trans. Green Commun. Netw. **6**, 1242–1257 (2021)
14. P. Mach, Z. Becvar, Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutor. **19**(3), 1628–1656 (2017)
15. Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017)
16. O. Munoz, A. Pascual-Iserte, J. Vidal, Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. IEEE Trans. Veh. Technol. **64**(10), 4738–4755 (2014)
17. P. Zhao, H. Tian, C. Qin, G. Nie, Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing. IEEE Access **5**, 11255–11268 (2017)
18. D. Huang, P. Wang, D. Niyato, A dynamic offloading algorithm for mobile computing. IEEE Trans. Wirel. Commun. **11**(6), 1991–1995 (2012)
19. S. Bi, L. Huang, H. Wang, Y.-J.A. Zhang, Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks. IEEE Trans. Wirel. Commun. **20**, 7519–7537 (2021)
20. B. Yang, X. Cao, J. Bassey, X. Li, L. Qian, Computation offloading in multi-access edge computing: a multi-task learning approach. IEEE Trans. Mob. Comput. **20**(9), 2745–2762 (2020)
21. T. Bai, C. Pan, C. Han, L. Hanzo, Reconfigurable intelligent surface aided mobile edge computing. IEEE Wirel. Commun. **28**(6), 80–86 (2021)
22. M. Di Renzo et al., Smart radio environments empowered by reconfigurable intelligent surfaces: how it works, state of research, and the road ahead. IEEE J. Sel. Areas Commun. **38**(11), 2450–2525 (2020)
23. E. Calvanese Strinati et al, Wireless environment as a service enabled by reconfigurable intelligent surfaces: the RISE-6G perspective, in *Proceedings of EUCNC 6G Summit* (Porto, Portugal, June 2021)
24. E.C. Strinati, G.C. Alexandropoulos, H. Wymeersch, B. Denis, V. Sciancalepore, R. D'Errico, A. Clemente, D.-T. Phan-Huy, E. De Carvalho, P. Popovski, Reconfigurable, intelligent, and sustainable wireless environments for 6g smart connectivity. IEEE Commun. Mag. **59**(10), 99–105 (2021)
25. M. Di Renzo et al., Smart radio environments empowered by reconfigurable AI meta-surfaces: an idea whose time has come. EURASIP J. Wirel. Commun. Netw. **1**, 1–20 (2019)
26. P. Mursia, V. Sciancalepore, A. Garcia-Saavedra, L. Cottatellucci, X. Costa-Pérez, D. Gesbert, RISMA: reconfigurable intelligent surfaces enabling beamforming for IOT massive access. IEEE J. Sel. Areas Commun. **39**, 1072–1085 (2020)
27. H. Zhang, B. Di, L. Song, Z. Han, Reconfigurable intelligent surfaces assisted communications with limited phase shifts: how many phase shifts are enough? IEEE Trans. Veh. Technol. **69**(4), 4498–4502 (2020)
28. C. Huang, A. Zappone, G.C. Alexandropoulos, M. Debbah, C. Yuen, Reconfigurable intelligent surfaces for energy efficiency in wireless communication. IEEE Trans. Wirel. Commun. **18**(8), 4157–4170 (2019)
29. Q. Wu, R. Zhang, Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming. IEEE Trans. Wirel. Commun. **18**(11), 5394–5409 (2019)
30. Y. Chen, M. Wen, E. Basar, Y.-C. Wu, L. Wang, W. Liu, Exploiting reconfigurable intelligent surfaces in edge caching: joint hybrid beamforming and content placement optimization. IEEE Trans. Wirel. Commun. **20**(12), 7799–7812 (2021)
31. T. Bai, C. Pan, Y. Deng, M. Elkashlan, A. Nallanathan, L. Hanzo, Latency minimization for intelligent reflecting surface aided mobile edge computing. IEEE J. Sel. Areas Commun. **38**(11), 2666–2682 (2020)
32. Z. Chu, P. Xiao, M. Shojafar, D. Mi, J. Mao, W. Hao, Intelligent reflecting surface assisted mobile edge computing for internet of things. IEEE Wirel. Commun. Lett. **10**, 619–623 (2020)
33. S. Huang, S. Wang, R. Wang, M. Wen, K. Huang, Reconfigurable intelligent surface assisted mobile edge computing with heterogeneous learning tasks. IEEE Trans. Cogn. Commun. Netw. **7**, 369–382 (2021)
34. X. Hu, C. Masouros, K.-K. Wong, Reconfigurable intelligent surface aided mobile edge computing: from optimization-based to location-only learning-based solutions. IEEE Trans. Commun. **69**, 3709–3725 (2021)
35. P. Di Lorenzo, M. Merluzzi, E.C. Strinati, Dynamic mobile edge computing empowered by reconfigurable intelligent surfaces, in *Proceedings of IEEE Workshop on Signal Processing Advances in Wireless Communications*, pp. 1–6 (2021)
36. E. Basar, I. Yildirim, SimRIS channel simulator for reconfigurable intelligent surface-empowered communication systems, in *2020 IEEE Latin-American Conference on Communications*, pp. 1–6 (2020). https://doi.org/10.1109/LATINCOM50620.2020.9282349
37. J.D.C. Little, A proof for the queuing formula: $l = \lambda w$. Oper. Res. **9**(3), 383–387 (1961)
38. T.D. Burd, R.W. Brodersen, Processor design for portable systems. J. VLSI Signal Process. Syst. **13**(2–3), 203–221 (1996). https://doi.org/10.1007/BF01130406
39. B. Debaillie, C. Desset, F. Louagie, A flexible and future-proof power model for cellular base stations, in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–7 (2015)
40. L.N. Ribeiro, S. Schwarz, M. Rupp, A.L. de Almeida, Energy efficiency of mmwave massive mimo precoding with low-resolution DACs. IEEE J. Sel. Top. Signal Process. **12**(2), 298–312 (2018)
41. R. Méndez-Rial, C. Rusu, N. González-Prelcic, A. Alkhateeb, R.W. Heath, Hybrid mimo architectures for millimeter wave communications: phase shifters or switches? IEEE Access **4**, 247–267 (2016)
42. M.J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems* (Morgan and Claypool, San Rafael, 2010)

43. D.P. Palomar, J.R. Fonollosa, Practical algorithms for a family of waterfilling solutions. IEEE Trans. Signal Process. **53**(2), 686–695 (2005)
44. G.C. Alexandropoulos, N. Shlezinger, P. Del Hougne, Reconfigurable intelligent surfaces for rich scattering wireless communications: recent experiments, challenges, and opportunities. IEEE Commun. Mag. **59**(6), 28–34 (2021)
45. R.L. Haupt, *Antenna Arrays: A Computational Approach* (IEEE Press, Piscataway, 2010)
46. A. Clemente, L. Dussopt, R. Sauleau, P. Potier, P. Pouliguen, Focal distance reduction of transmit-array antennas using multiple feeds. IEEE Antennas Wirel. Propag. Lett. **11**, 1311–1314 (2012). https://doi.org/10.1109/LAWP.2012.2227105