Online Convex Optimization for Efficient and Robust Inter-Slice Radio Resource Management

Tianyu Wang, *Member, IEEE*, and Shaowei Wang, *Senior Member, IEEE*

***Abstract*—Radio access network (RAN) slicing is one of the key technologies in 5G and beyond mobile networks, where multiple logical subnets, i.e., RAN slices, are allowed to run on top of the same physical infrastructure so as to provide slice-specific services. Due to the dynamic environments of wireless networks and the diverse requirements of RAN slices, inter-slice radio resource management (IS-RRM) has become a highly challenging task in RAN slicing. In this paper, we propose a novel online convex optimization (OCO) framework for IS-RRM, which directly learns the instant resource allocation from the data revealed by previous allocations, such that sophisticated modeling and parameterization can be avoided in highly complicated and dynamic wireless environments. Specifically, an online IS-RRM scheme that employs multiple expert-algorithms running in parallel is proposed to keep track of the environmental changes and adjust the resource allocation accordingly. Both theoretical analysis and simulation results show that our proposed scheme can guarantee long-term performance comparable to the optimal strategies given in hindsight.**

***Index Terms*—Network slicing, online convex optimization, radio access network, radio resource management.**

I. Introduction

5G and beyond mobile networks are expected to support a variety of vertical industries, ranging from human-centric multimedia services with high peak data rates to vehicular communications with ultra-reliable and low-latency requirements, and logistics applications with massive IoT connections. Due to the highly diversified requirements in terms of throughput, latency, reliability and availability, the conventional "one-size-fits-all" network architecture that provides homogeneous treatment to all services becomes an unacceptable compromise in both the technical and economic aspects. Instead, network slicing is envisaged as a promising technology that allows multiple logical networks, i.e., network slices, to run on top of the same physical infrastructure to provide slice-specific services [1, 2].

Network slices are treated as independent logical networks with an end-to-end performance guarantee, involving both the radio access network (RAN) and the core network components. Compared to core network slicing which can be achieved by powerful computing and caching components, RAN slicing is more challenging due to the limited and unscalable radio resources, the highly diversified and stringent requirements of RAN slices and the dynamic environments of wireless networks. Recently, RAN slicing has attracted considerable interest from both academia and industry [3, 4]. Specifically, it has been agreed by 3GPP that 5G should deploy a slice-aware RAN, where different L1/L2 configurations must be provided to handle slice-specific traffic [5].

However, it has not come to any final agreement on how the RAN should be specifically sliced. Instead, a variety of high-level architectures and solutions are proposed in the literature [6–8]. These solutions define RAN slices with different granularity according to their service types (e.g., autonomous driving and environmental monitoring), technical requirements (e.g., bandwidth and latency) and business companies (e.g., different vertical companies), and provide multiplexing options with different levels of infrastructure and spectrum reuse, e.g., the stand-alone option for safety-critical use cases like automatic driving, and the sharing option with common physical and MAC layers for machine type services with massive IoT connections.

Despite the differences, it is widely agreed that multi-slice radio resource management (RRM) is of critical importance for the success of these high-level solutions [7]. Due to the complexity of slice-aware RANs, the conventional MAC scheduler that directly deals with radio resources for each individual user in every transmission time interval (TTI) has become a multi-objective optimization problem that is too complicated to solve in practical deployments [6]. Instead, multi-slice RRM is decomposed into inter-slice and intra-slice RRM, which can be performed separately across different time scales [6, 8]. Intra-slice RRM, which is responsible for allocating slice-dedicated resources to the corresponding slice users, focuses on the packet level performance and can be realized by using conventional MAC schedulers operating in every TTI. Inter-slice RRM, which is abbreviated as IS-RRM in this paper, is responsible for assigning radio resources to each slice for a predefined allocation window. It focuses on the satisfaction of service level agreements (SLAs) and has become a new technical issue.

The challenges of IS-RRM come from two sides. On the one hand, radio resources such as frequency spectrum, transmission power, access points and signaling resources are limited and should be efficiently utilized. On the other hand, RAN slices are envisioned as independent networks that are logically "isolated" from each other, which implies that the SLA performance of a slice should be robust to the changes of others. In general, the requirements of these two sides are

Manuscript received October 28, 2020; revised March 4, 2021 and May 27, 2021; accepted May 31, 2021. This work was partially supported by the National Natural Science Foundation of China under Grants 61801208, 61931023, and U1936202. The associate editor coordinating the review of this article and approving it for publication was Xavier Costa-Prez. *(Corresponding author: Shaowei Wang.)*

The authors are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (email: tianyu.alex.wang@nju.edu.cn, wangsw@nju.edu.cn).

contradictory, which fundamentally determines that IS-RRM is a tradeoff between the utilization efficiency of radio resources and the slice robustness to network dynamics. In the literature, IS-RRM schemes are proposed mainly based on three mathematical frameworks, i.e., classic optimization technique [9–19], reinforcement learning [20–23] and game theory [24–27]. However, each framework has its own drawbacks considering the inherent characteristics of IS-RRM.

The classic optimization approach formulates an optimization problem, in which the objective function quantifies the relationship between the considered key performance indicator (KPI) and the radio resource allocation, and meanwhile the constraints represent the radio resource limitations [9–19]. Analytical mathematical models are usually adopted to simplify the problem formulation, e.g., the packet arrival is typically modeled as a Poisson process. However, despite its mathematical elegance, the optimization approach suffers from the fundamental limitation of modeling accuracy, leading to great difficulties of model determination and parameterization in practical wireless networks. Besides, some network behaviors are highly complex, e.g., slice-specific packet schedulers with customized scheduling policies, which cannot be appropriately described by using conventional mathematical models with only a few parameters.

To avoid the difficulty of establishing sophisticated mathematical models for network behaviors, recent works resort to the reinforcement learning technology, in which the IS-RRM problem is treated as a Markov decision process and the optimal policy is achieved via the interaction between the IS-RRM engine and the network environment [20–23]. In addition, to avoid the curse of dimensionality brought by the large action space, deep neural networks are usually incorporated, which makes them deep reinforcement learning (DRL) based methods. For online DRL methods, the policy is periodically updated with streaming data collected by the online interaction itself, which, however, may lead to unacceptable performance variations due to random explorations. For offline DRL methods, which intend to learn the optimal policy from previously collected offline data without any additional online interaction, the major technical challenge is to handle the distribution shift, i.e., the neural network may be trained under one distribution while evaluated on a different distribution due to possible changes of network characteristics.

Game theory is a suitable framework to reflect the conflict of interest among the business companies behind each slice [24–27]. From a game theory point of view, the considered IS-RRM problem can be treated as a repetitive game in which multiple selfish players, i.e. RAN slices, periodically compete for the limited radio resources to maximize their individual utilities. The outcome is represented by a static strategy combination of all slices, referred to as equilibrium, in which no slice can benefit by simply changing its own strategy. However, the major drawback of game theory methods is that the outcome equilibrium does not necessarily lead to the optimal solution from the operator's point of view, which is called the price of anarchy. In addition, to converge to the equilibrium, game theory methods usually require the players to iteratively perform a large number of strategy updates, which results in high computation and communication burdens.

In this article, we consider the IS-RRM in a single-cell RAN, where online convex optimization (OCO) is proposed for the dynamic assignment of downlink bandwidth resources [28]. Compared with the classic optimization methods, the OCO formulates an online process that can gradually learn the network dynamics from the data revealed sequentially in previous allocations, which avoids the difficulty of modeling and achieves a low computational complexity. Compared with online DRL, the OCO avoids the blind exploration in action space by exploiting the derivatives of well-designed loss functions and then assures the variation of online network performance. Compared with offline DRL, the OCO avoids the distributional shift by fitting time-varying loss functions with online data. Compared with game theory, the OCO avoids the price of anarchy by treating the IS-RRM as a centralized problem. Our contributions are summarized as follows:

- We formulate IS-RRM as an OCO problem, for which a constant projection space is established to represent the bandwidth limitation and the loss function in each allocation window is carefully designed to represent the dynamic SLA performance. The proposed OCO framework can avoid the drawbacks of conventional methods in modeling difficulty, computational complexity, performance guarantee and deployment flexibility.
- We develop an online IS-RRM algorithm to address the formulated optimization task, which dynamically assigns bandwidth resources to each RAN slice in each allocation window. The proposed algorithm is a model-free scheme that can directly learn from the experience data so as to adjust the resource allocation accordingly. Specifically, it employs multiple expert-algorithms running in parallel and dynamically combines their decisions to keep track of the environmental changes. The computational complexity is logarithmic to both the slice number and the considered time horizon, which is significantly lower than other aforementioned methods.
- We provide a detailed theoretical analysis of the proposed IS-RRM scheme, which shows that our proposed algorithm can achieve a sublinear "adaptive regret", i.e., the cumulative performance gap between the proposed OCO algorithm and the optimal strategy in hindsight grows sublinearly with time. In addition, we show that this gap has a problem-dependent bound, which increases sublinearly with the average and burst traffic rates.
- We conduct a series of experiments to verify the proposed IS-RRM scheme, which confirms the theoretical analysis as well as the effectiveness and robustness of the proposed algorithm.

The remainder of this article is organized as follows. In Section II, we provide a literature review of the existing methods. In Section III, the system model is presented, and in Section IV, we formulate IS-RRM as an OCO problem and define the corresponding loss functions. In Section V, we propose an online IS-RRM algorithm and provide theoretical analysis on the adaptive regret. Simulation results are given in Section VI, and we conclude the article in Section VII.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3087127, IEEE Transactions on Communications

II. Related Work

In optimization-based IS-RRM methods, network virtualization substrate is one of the earliest works, which operates at MAC-frame granularity to decouple the slice scheduling problem from the packet scheduling problem [10, 11]. To address SLAs with finer time granularity, mobile traffic forecasting technology is introduced to predict the actual footprint of each particular slice [12], and short-term KPIs are considered by assumptively transforming the SLAs into the requirements of physical radio resources in each allocation period [9, 13]. To achieve the tradeoff between spectrum efficiency and slice isolation in a large time granularity, the allocation window is extended into hours, where the SLA requirements are uniformly described by using a guaranteed demand parameter and an overbooking penalty parameter [17]. To reduce the computational complexity of the optimization approach, simplification techniques and heuristic algorithms are widely considered in the literature [9, 12, 13, 16]. Also, for multi-cell scenarios, the users at the cell edge can be served by multiple base stations in proximity using coordinated multi-point transmissions and beamforming, which further complicates the radio resource allocation problem [18, 19].

In DRL-based IS-RRM methods, a DRL-based dynamic pricing mechanism is proposed to provide incentives for the slice owners to share the spectrum resources in an efficient manner [20]. DRL-based bandwidth allocation algorithms are also proposed [21, 22], where discrete normalized advantage functions and generative adversarial networks are introduced to accelerate the convergence rate and improve the approximation accuracy, respectively. A long short-term memory method is incorporated into the reinforcement learning framework to further extract and exploit the user mobility patterns [23].

In game theory-based IS-RRM methods, a joint user association and bandwidth allocation problem is formulated as a congestion game, for which a distributed semi-online algorithm is proposed to quickly converge to an equilibrium that meets the SLAs [24]. The distributed admission control problem is formulated as a traffic shaping game, for which a share constrained proportionally fair algorithm is proposed to achieve an equilibrium that is robust in terms of bit transmission delay [25]. Network slicing games with elastic and inelastic requirements are also analyzed [26, 27].

The proposed OCO is a novel framework that is fundamentally different from the existing methods, which formulates IS-RRM as a model-free and online process by using dynamic loss functions defined in each allocation window. OCO has been recently considered an emerging methodology for dynamic decision-making tasks, such as wireless movement tracking, data center management and edge computing [29–38]. As far as the authors have known, this is the first article that exploits the OCO framework for IS-RRM. We note that the proposed online IS-RRM algorithm is based on the state-of-the-art online learning algorithm with multiple expert-algorithms running in parallel, which can be further extended to more complicated scenarios with multiple KPIs and multiple radio access points.

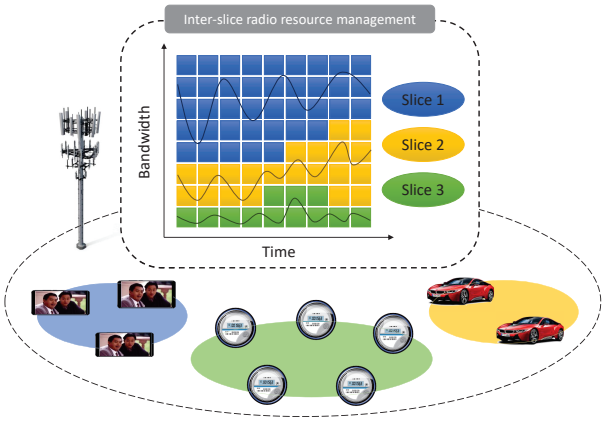


Fig. 1: Inter-slice radio resource management.

III. System Model

As illustrated in Fig. 1, we consider a single-cell RAN with N slices, where each slice provides a customized network service for multiple user equipments (UEs). The IS-RRM is periodically performed to allocate bandwidth resources to each slice according to the network dynamics. The bandwidth resources are organized into W resource blocks (RBs) in each TTI, and the length of an allocation window is L TTIs. For any allocation window t, or round t, the corresponding RB allocation is denoted by $\boldsymbol{w}_t = (w_{t,1}, \ldots, w_{t,N})$, where $w_{t,n}$ is the number of dedicated RBs of slice n. Thus, for any round t within a time horizon $[1, T]$, we have $\boldsymbol{w}_t \in \mathcal{W}_{\mathbb{N}}$, where

$$\mathcal{W}_{\mathbb{N}} = \left\{ \boldsymbol{x} \in \mathbb{N}^N \mid \sum_{n=1}^{N} x_n \leq W \right\}. \tag{1}$$

For any UE k of slice n at round t, we denote by $\mathcal{H}_{t,n,k}$ the channel state information of the WL RBs in the corresponding allocation window, by $\mathcal{P}_{t,n,k}$ the set of packets that arrive at the traffic queue during the allocation window, and by $\mathcal{Q}_{t,n,k}$ the set of packets that are already buffered in the traffic queue at the beginning of the allocation window. For simplicity, we set $\mathcal{H}_{t,n} = \{\mathcal{H}_{t,n,k}\}_k$, $\mathcal{P}_{t,n} = \{\mathcal{P}_{t,n,k}\}_k$ and $\mathcal{Q}_{t,n} = \{\mathcal{Q}_{t,n,k}\}_k$, and set $\mathcal{H}_t = \{\mathcal{H}_{t,n}\}_n$, $\mathcal{P}_t = \{\mathcal{P}_{t,n}\}_n$ and $\mathcal{Q}_t = \{\mathcal{Q}_{t,n}\}_n$. Specifically, $\mathcal{P}_{[r,s]} = \bigcup_{t=r}^{s} \mathcal{P}_t$ denotes the set of packets that arrive at the traffic queues during the time interval between round t and round s.

A. Two-Level Scheduler

As shown in Fig. 2, a two-level MAC scheduler is applied [6, 39], which takes the RB allocation $\boldsymbol{w}_t$ given by the IS-RRM as its input constraining the number of dedicated RBs of each slice and outputs the scheduling decisions for all RBs in each of the corresponding L TTIs. Specifically, for the RAN shown in Fig. 1, it involves three slice-specific schedulers as well as a common scheduler for all slices. Each slice-specific scheduler is responsible for provisioning packet-level performance guarantee, where a highly customized scheduling algorithm is adopted to assign virtual RBs (vRBs) to the corresponding UEs. We denote by $\mathcal{V}_n$ the slice-specific scheduling

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3087127, IEEE Transactions on Communications

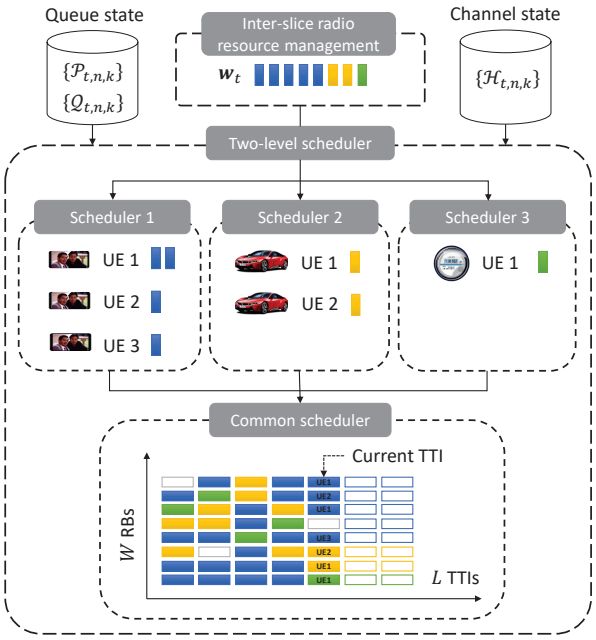


Fig. 2: Two-level scheduler for multi-slice packet scheduling.

algorithm of slice n. We assume that the size of a vRB is aligned with the physical RB (pRB), and thus the maximal number of vRBs that can be scheduled by $\mathcal{V}_n$ at round t is specifically given by $w_{t,n}$.

The common scheduler is responsible for translating the UE-vRB assignments given by each slice-specific scheduler into a common UE-pRB assignment. Here, each pRB is assigned to the UE that achieves the maximum throughput in the current TTI so as to maximize the inter-slice multi-user gain. Note that this scheduling strategy does not lead to the starvation of UEs in poor channel conditions since the number of pRBs assigned to each UE is guaranteed by the slice-specific scheduler. Compared with the conventional schedulers that jointly dispatch all packets in one pass, the two-level scheduler utilizes multiple slice-specific schedulers in parallel, which decouples the multi-dimensional scheduling task into multiple single-dimensional scheduling ones and achieves both high flexibility and low computational complexity.

B. Service Level Agreement

In practical networks, the network operator needs to carefully study the SLAs of all potential slices and balance their performance based on technical and business concerns. For simplicity without loss of generality, we assume the SLAs are represented by the packet loss ratio (PLR) under a strict packet delay budget. Specifically, for any slice n, the slice-specific scheduler $\mathcal{V}_n$ can schedule packets within the maximum delay d_n and no retransmission is allowed if the packets are erroneously received. The PLR is defined as the ratio of the number of lost packets to the total number of packets that arrive at the base station. In any round t, the number of arriving packets is given by $|\mathcal{P}_{t,n}|$ and the number of lost packets is denoted by $C_{t,n}$. Note that $C_{t,n}$ depends on multiple influence factors, including the traffic state $\mathcal{P}_{t,n}$ and $\mathcal{Q}_{t,n}$, the channel state $\mathcal{H}_{t,n}$, the available bandwidth resources $w_{t,n}$, as well as the slice-specific scheduling algorithm $\mathcal{V}_n$. Thus, we can rewrite it as $C_{t,n}(\mathcal{P}_{t,n}, \mathcal{Q}_{t,n}, \mathcal{H}_{t,n}, w_{t,n}|\mathcal{V}_n)$ and the PLR of slice n after round t is given by

$$r_{t,n} = \frac{\sum_{s=1}^{t} C_{s,n}(\mathcal{P}_{s,n}, \mathcal{Q}_{s,n}, \mathcal{H}_{s,n}, w_{s,n} \mid \mathcal{V}_n)}{\sum_{s=1}^{t} |\mathcal{P}_{s,n}|}. \quad (2)$$

For any slice n, the service requirement is represented by a target PLR r_n. Thus, the SLA performance of slice n can be defined as

$$g_n(t) = \begin{cases} 0, & \text{if } r_{t,n} < r_n \\ u(r_{t,n} - r_n), & \text{otherwise,} \end{cases} \quad (3)$$

where $u(\cdot)$ is a positive and superlinearly increasing function defined in $[0, +\infty)$, i.e., $u(\cdot)$ eventually grows faster than any linear function.

C. Problem Formulation

The considered IS-RRM can be formulated as a multi-objective programming problem that aims to optimize the SLA performance of all slices, which is formally written as

$$\min_{\{\boldsymbol{w}_t \in \mathcal{W}_{\mathbb{N}}\}_t} \left(\sum_{t=1}^{\infty} g_1(t), \sum_{t=1}^{\infty} g_2(t), \ldots, \sum_{t=1}^{\infty} g_N(t) \right). \quad (4)$$

Typically, there does not exist a feasible allocation sequence that can simultaneously minimize $\sum_{t=1}^{\infty} g_n(t)$ for all $n \in \{1, 2, \ldots, N\}$. Instead, Pareto optimal solutions are usually considered [40], which are defined as feasible solutions that cannot be improved in any of the objectives $\sum_{t=1}^{\infty} g_n(t)$ without degrading at least one of the other objectives $\sum_{t=1}^{\infty} g_{n'}(t), n' \neq n$.

We note that there are two major drawbacks of the above formulation. The first drawback is that considering the complexity of the slice-specific scheduling algorithms (e.g., $\mathcal{V}_n$), there may not exist an analytical function that can quantify the mathematical relationship between the objective $\sum_{t=1}^{\infty} g_n(t)$ and the allocation sequence $\{\boldsymbol{w}_t\}_t$, which makes problem (4) extremely difficult to solve. The second drawback is that considering the small granularity of IS-RRM, the involved parameters (e.g., $\mathcal{P}_{t,n}, \mathcal{Q}_{t,n}, \mathcal{H}_{t,n}$) can be highly dynamic and hard to predict with presumed statistical models, which may cause severe performance degradation of the outcome solutions. In the next section, we introduce the OCO framework to reformulate the IS-RRM problem, which avoids the drawbacks of multi-objective programming.

IV. Problem Reformulation

In this section, we consider the IS-RRM from an online learning point of view, where the instant RB allocation is learned online by using the data revealed from previous rounds, rather than being calculated or optimized via comprehensive models given beforehand. This online process allows the base station to learn from the experience and provide better RB allocations adaptively as more cases are observed.

A. OCO Preliminaries

In OCO, an online learner iteratively makes decisions to minimize its cumulative loss. At each round t, the decision is denoted as $\boldsymbol{x}_t \in \mathcal{X}$, and the outcomes associated with that decision are unknown to the learner. After committing to decision $\boldsymbol{x}_t$, a loss function $f_t \in \mathcal{F} : \mathcal{X} \to \mathbb{R}$ is revealed, and the learner suffers from a loss $f_t(\boldsymbol{x}_t)$. The decision set is a convex set in Euclidean space, i.e., $\mathcal{X} \subseteq \mathbb{R}^N$, and the loss functions $\{f_t\}_t$ are bounded convex functions over $\mathcal{X}$.

Since the loss functions can only be obtained in hindsight, it is usually unlikely to minimize the actual cumulative loss. Instead, an appropriate performance metric is the difference between the cumulative loss incurred by the learner and that of the optimal decision in hindsight, which is referred to as *regret* [28]. Formally, for any horizon T, the regret of an OCO algorithm $\mathcal{A}$ is defined as

$$\textit{Regret}_{\mathcal{A}}(T) = \sup_{\{f_t \in \mathcal{F}\}_t} \left\{ \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x}) \right\}. \quad (5)$$

Thus, the regret is defined as the loss difference in the worst case for all possible loss functions. An algorithm is called a low-regret algorithm if its regret is sublinear as a function of T, i.e., $\textit{Regret}_{\mathcal{A}}(T) = o(T)$. It implies that on the average low-regret algorithms can perform as well as the optimal decision in hindsight as T goes to infinity. Classical low-regret algorithms include online gradient decent and online Newton step, which utilize the first and second order derivatives of the loss functions, respectively, to ensure bounded regrets in an arbitrary environment [28].

However, in real-world applications with highly dynamic loss functions, e.g., for the considered IS-RRM problem, the best solution in hindsight is usually a dynamic decision sequence, which may greatly outperform the benchmark static decision given in (5). Thus, the classic regret is no longer a suitable performance metric. To cope with highly dynamic environments, the notion of *adaptive regret* has been proposed [41], which is defined as the worst-case regret in any contiguous interval with a predefined length. Formally, for any horizon T and interval length τ, the adaptive regret of an OCO algorithm $\mathcal{A}$ is defined as

$$\textit{A-Regret}_{\mathcal{A}}(T, \tau) = \max_{[\tau] \subseteq [T]} \textit{Regret}_{\mathcal{A}}([\tau]), \quad (6)$$

where $[\tau]$ represents a contiguous time interval with length τ in horizon T. An algorithm is called a low-adaptive-regret algorithm if its adaptive regret is sublinear as a function of both T and τ, i.e., $\textit{A-Regret}_{\mathcal{A}}(T, \tau) = o(T)o(\tau)$. It implies that on average low-adaptive-regret algorithms can perform as well as the optimal decisions in hindsight in any interval $[\tau]$ as T and τ go to infinity. In this paper, we use adaptive regret to provide a theoretical performance analysis of our proposed online IS-RRM algorithm.

B. IS-RRM As OCO

For the considered IS-RRM problem, the base station can be seen as a learner that periodically decides the RB allocation $\boldsymbol{w}_t$ in each round t so as to optimize the overall SLA performance in total T rounds. The feasible set of RB allocations $\mathcal{W}_{\mathbb{N}}$ is relaxed by using its convex hull, which is a standard N-simplex scaled by W, given by

$$\mathcal{W} = \left\{ \boldsymbol{x} \in \mathbb{R}^N \mid \sum_{n=1}^{N} x_n \leq W \text{ and } x_n \geq 0, \forall n \right\}. \quad (7)$$

Thus, we have $\boldsymbol{w}_t \in \mathcal{W}$ for all $t \in \{1, 2, \ldots, T\}$. Note that the constraint in (4) can still be satisfied by rounding the coordinates of $\boldsymbol{w}_t$ before it is committed in round t.

The loss function can be designed based on the PLR performance of each packet in each round. Specifically, we consider an arbitrary packet p that arrives in the queue of slice n_p at the beginning of the l_p^{in}-th TTI of round t_p^{in} and leaves at the end of the l_p^{out}-th TTI of round t_p^{out}. The packet size is B_p. For simplicity, we denote the l-th TTI of round t by a tuple (t, l), and the time period between any two TTIs (t, l) and (t', l') is given by

$$\begin{aligned} \mathcal{T}[(t,l),(t',l')] = &\{(s,m) \mid s = t \ \&\ s < t', m \in [l, L]\} \cup \\ &\{(s,m) \mid s = t' \ \&\ s > t, m \in [1, l']\} \cup \\ &\{(s,m) \mid s \in (t, t'), m \in [1, L]\} \cup \\ &\{(s,m) \mid s = t \ \&\ s = t', m \in [l, l']\}. \end{aligned} \quad (8)$$

Specifically, $\mathcal{T}_p = \mathcal{T}[(t_p^{in}, l_p^{in}), (t_p^{out}, l_p^{out})]$ denotes the lifetime of packet p, and $\mathcal{T}_p(t, l) = \mathcal{T}[(t_p^{in}, l_p^{in}), (t, l)]$ denotes the experienced lifetime of packet p at TTI (t, l).

For any TTI $(t, l) \in \mathcal{T}_p$, we denote by $B_{p,t,l}$ the amount of packet p's data bits that are scheduled by the MAC scheduler. Packet p is fulled scheduled if $\sum_{(t,l) \in \mathcal{T}_p} B_{p,t,l} = B_p$ and it is dropped due to the lack of radio resources if $\sum_{(t,l) \in \mathcal{T}_p} B_{p,t,l} < B_p$. For any round $t \in [t_p^{in}, t_p^{out}]$, we define the cumulative loss of packet p as

$$\begin{aligned} \Theta_{p,t} = &\sum_{(s,m) \in \mathcal{T}_p(t,L)} B_{p,s,m} \left(\frac{d_{p,s,m}}{d_{n_p}} \right)^2 + \Bigg(B_p - \\ &\sum_{(s,m) \in \mathcal{T}_p(t,L)} B_{p,s,m} \Bigg) \left(\frac{d_{p,t,L}}{d_{n_p}} \right)^2, \end{aligned} \quad (9)$$

where $d_{p,s,m} = |\mathcal{T}_p(s, m)|$ represents the experienced delay of packet p at TTI (s, m), and d_{n_p} is the maximum delay of packet p, or equally the packet delay budget of slice n_p. Note that $d_{p,s,m} \leq d_{n_p}$ for all TTIs $(s, m) \in \mathcal{T}_p$, and $d_{p,s,m} = d_{n_p}$ if $(s, m) \in \mathcal{T}[(t_p^{out}, l_p^{out}), (t_p^{out}, L)]$. For simplicity, we set $\Theta_{p,t} = 0$ for all $t < t_p^{in}$ and $\Theta_{p,t} = \Theta_{p,t_p^{out}}$ for all $t > t_p^{out}$.

As seen in (9), the cumulative loss of packet p consists of two parts, indicating the loss of transmitted and buffered bits, respectively. For each part, the per-bit loss is defined to be proportional to the square of experienced delay normalized by the maximum delay. Thus, $\Theta_{p,t}$ can reflect the PLR performance of packet p by indicating how close it is about to be dropped in the upper layer. Specifically, we have $\Theta_{p,t} \in (0, B_p]$, and it is a strictly increasing function of t.

Given the cumulative loss of each packet as defined in (9), the loss of slice n in round t is then defined as

$$\theta_{t,n} = \sum_{p \in \mathcal{P}_{t,n} \cup \mathcal{Q}_{t,n}} \left(\Theta_{p,t} - \Theta_{p,t-1} \right), \quad (10)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3087127, IEEE Transactions on Communications

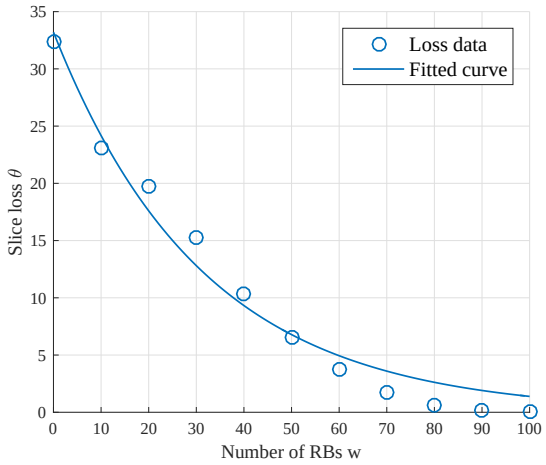


Fig. 3: Illustration of slice loss θ as a function of RB number w. The fitted curve is given by $\theta = 33.21 \times \exp\{-0.03176w\}$ and the coefficient of determination is 0.9773. The slice has an average data rate of 30 Mbps and a packet delay budget of 100 ms. The spectrum efficiency is between 2 bps/Hz and 5 bps/Hz and the allocation window is 10 ms.

which represents the sum of the marginal increase of $\Theta_{p,t}$ for each packet p of slice n in round t. Note that $\theta_{t,n} \in (0, B_{n,t}]$ where $B_{n,t} = \sum_{p \in \mathcal{P}_{t,n} \cup \mathcal{Q}_{t,n}} B_p$ is the total amount of data bits that are involved in the scheduling process.

For any given number of RBs $w_{t,n}$, the loss $\theta_{t,n}$ can be obtained in hindsight after the network parameters $\mathcal{P}_{t,n}, \mathcal{Q}_{t,n}$ and $\mathcal{H}_{t,n}$ are observed in the transmission of round t. Intuitively, $\theta_{t,n}$ is a positive function that decreases with $w_{t,n}$ and approaches to 0 as $w_{t,n}$ goes to infinity. In Fig. 3, we illustrate $\theta_{t,n}$ as a function of $w_{t,n}$ in a typical network setting. We can see that a negative exponential function can be well fitted to the hindsight data. Therefore, the loss of slice n in round t can be approximated by a negative exponential function of $w_{t,n}$, given by

$$\tilde{\theta}_{t,n}(w_{t,n}) = a_{t,n} \cdot \exp\left\{b_{t,n} \cdot w_{t,n}\right\}, \tag{11}$$

where $a_{t,n} > 0$ and $b_{t,n} < 0$ are the coefficients of the fitted curve. We note that $\tilde{\theta}_{t,n}$ is a strictly decreasing and convex function of $w_{t,n}$, and $0 < \tilde{\theta}_{t,n}(w_{t,n}) \leq a_{t,n}$. Specifically, we denote a^* as an upper bound of $a_{t,n}$, i.e., $a_{t,n} < a^*, \forall t, n$.

We can obtain the parameters $a_{t,n}$ and $b_{t,n}$ by using curve fitting tools, which simply solves the following problem

$$\min_{a_{t,n}, b_{t,n}} \sum_{w=1}^{W} \left(\tilde{\theta}_{t,n}(w|a_{t,n}, b_{t,n}) - \theta_{t,n}(w)\right)^2. \tag{12}$$

In addition, to reduce the online computational complexity, we can deploy a deep neural network to approximate the loss function, which takes the hindsight data of allocation window t as its input and outputs the approximated $a_{t,n}$ and $b_{t,n}$.

The overall PLR performance is then formulated by introducing a priori weight $\alpha_{t,n}$ for each slice n. Specifically, the weight factor is defined as

$$\alpha_{t,n} = \frac{r_{t,n}}{r_n}, \tag{13}$$

where $r_{t,n}$ is the current PLR of slice n and r_n is the target PLR. Specifically, we denote r^* as the minimum value of all target PLRs, i.e., $r^* = \min\{r_1, r_2, \ldots, r_N\}$. Therefore, the loss function in round t is defined as

$$F_t(\boldsymbol{w}_t) = \frac{1}{M} \sum_{n=1}^{N} \alpha_{t,n} \tilde{\theta}_{t,n}(w_{t,n}), \tag{14}$$

where $M = Na^*/r^*$ is a normalization factor that ensures $F_t(\boldsymbol{w}_t) \leq 1$. Since $\tilde{\theta}_{t,n}$ is a strictly monotonic and convex function of $w_{t,n}$ and $\alpha_{t,n} \geq 0$, we have F_t is a convex function of $\mathbf{w}_t$. Therefore, the IS-RRM can be formulated as an OCO problem, where the convex RB allocation set is $\mathcal{W}$ and the convex loss function of round t is given by $F_t(\cdot)$. In the next section, a low-adaptive-regret algorithm is proposed for the considered IS-RRM problem.

V. Online IS-RRM Algorithm

In this section, we first introduce the strongly adaptive algorithm for convex and smooth functions (SACS) [42], which has proven to be able to achieve a sublinear adaptive regret with a problem-dependent bound. Then we propose the online IS-RRM algorithm based on SACS and analyze its adaptive regret for the considered IS-RRM problem.

A. Online IS-RRM Based on SACS

The SACS algorithm contains three parts: 1) An expert-algorithm, which is able to minimize the classical regret within a given time interval; 2) A set of intervals, each of which is associated with an instance of the expert-algorithm running in that interval; 3) A meta-algorithm, which combines the decisions of active experts in each round. Thus, the SACS algorithm can be seen as an ensemble learning method that runs multiple expert-algorithms in parallel in different time intervals and dynamically combines their decisions by updating their weights in each round. To utilize the SACS algorithm, three assumptions must be satisfied.

Assumption 1. *The diameter of the feasible domain $\mathcal{X}$ is bounded by D, i.e.,*

$$\max_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\| \leq D. \tag{15}$$

Assumption 2. *The value of each loss function belongs to $[0, 1]$, i.e.,*

$$0 \leq f_t(\boldsymbol{x}) \leq 1, \tag{16}$$

for all $\boldsymbol{x} \in \mathcal{X}$ and $t \in \{1, 2, \ldots, T\}$.

Assumption 3. *All loss functions are H-smooth over $\mathcal{X}$, i.e.,*

$$\|\nabla f_t(\boldsymbol{x}) - \nabla f_t(\boldsymbol{x}')\| \leq H \|\boldsymbol{x} - \boldsymbol{x}'\|, \tag{17}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and $t \in \{1, 2, \ldots, T\}$.

For the considered IS-RRM problem, the feasible domain $\mathcal{W}$ given in (7) is a standard N-simplex scaled by W, whose diameter is given by $D = \sqrt{2}W$. The loss functions $F_t(\boldsymbol{w})$ defined in (14) belong to $[0, 1]$, as we have noted. Thus, the considered IS-RRM problem satisfies the first two

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3087127, IEEE Transactions on Communications

assumptions. Note that $F_t(\boldsymbol{w})$ is twice differentiable, and its Hessian is a diagonal matrix given by

$$\nabla^2 F_t(\boldsymbol{w}) = \begin{bmatrix} H_{t,1}e^{b_{t,1}w_1} & & \\ & \ddots & \\ & & H_{t,N}e^{b_{t,N}w_N} \end{bmatrix},$$

where $H_{t,n} = \alpha_{t,n}a_{t,n}b_{t,n}^2/M$ is always bounded. We denote by H as an upper bound of $H_{t,n}$, i.e., $H \geq H_{t,n}, \forall t, n$. Thus, we have $H\boldsymbol{I}_N - \nabla^2 F_t(\boldsymbol{w})$ is positive semidefinite, or equally, $F_t(\boldsymbol{w})$ is H-smooth. Therefore, the considered IS-RRM problem satisfies all three assumptions, and the SACS algorithm can be applied safely.

We first introduce the expert-algorithm, scale-free online gradient descent (SOGD), which can automatically utilize the smoothness condition to improve the regret performance. For any round t, the set of active experts is denoted by $\mathcal{K}_t$. We consider an arbitrary expert $k \in \mathcal{K}_t$ that is activated at round t_k^s and deactivated after round t_k^e. The RB allocation determined by expert k, denoted by $\boldsymbol{w}_t^{(k)}$, is updated by

$$\boldsymbol{w}_t^{(k)} = \Pi_{\mathcal{W}}\left[\boldsymbol{w}_{t-1}^{(k)} - \eta_{t-1,k}\nabla F_{t-1}(\boldsymbol{w}_{t-1}^{(k)})\right], \quad (18)$$

where $\Pi_{\mathcal{W}}$ is the operation that projects any point in $\mathbb{R}^N$ to the closet point inside convex set $\mathcal{W}$, i.e.,

$$\Pi_{\mathcal{W}}(\boldsymbol{w'}) \coloneqq \arg\min_{\boldsymbol{w}\in\mathcal{W}} \|\boldsymbol{w} - \boldsymbol{w'}\|, \quad (19)$$

and $\eta_{t-1,k}$ is a time-varying step size given by

$$\eta_{t-1,k} = \frac{D/\sqrt{2}}{\sqrt{\delta + \sum_{s=t_k^s}^{t-1}\left\|\nabla F_s(\boldsymbol{w}_s^{(k)})\right\|^2}}, \quad (20)$$

where $\delta > 0$ is a small number that is introduced to avoid being divided by 0. The initial RB allocation $\boldsymbol{w}_{t_k^s}^{(k)} \in \mathcal{W}$ can be arbitrarily determined. Here, we adopt the average allocation $\boldsymbol{w}_{t_k^s}^{(k)} = (W/N, W/N, \ldots, W/N)$ for all experts.

The active intervals of each SOGD expert are determined online. First, a sequence of *markers* is generated by repeatedly restarting an SOGD instance when its cumulative loss exceeds a certain threshold C, given by

$$C = 20HD^2 + 2D\sqrt{2\delta}. \quad (21)$$

Specifically, the SOGD instance starts at round $s_1 = 1$ and keeps running until its cumulative loss $\sum_{t=s_1}^{s_1+\tau_1-1} F_t(\boldsymbol{w}_t^{(k)}) > C$ after τ_1 rounds. Then, the SOGD instance is restarted at round $s_2 = s_1 + \tau_1$. Repeating this process, we can generate online a sequence of markers $s_1, s_2, \ldots, s_S$ with $S \leq T$. Given the markers, the set of SOGD intervals are given by

$$\mathcal{I} = \bigcup_{k\in\mathbb{N}\cup\{0\}} \mathcal{I}_k, \quad (22)$$

where

$$\mathcal{I}_k = \left\{\left[s_{i\cdot 2^k}, s_{(i+1)\cdot 2^k} - 1\right], i \in \mathbb{N}\right\}. \quad (23)$$

In Fig. 4, the set of SOGD intervals is illustrated for a sequence of makers $(1, 3, 5, 8, 9, 10, 12, 13, 15, ...)$. In the first 16 rounds, 4 groups of intervals are generated, i.e., $\mathcal{I}_0 = \{[1,2],[3,4],[5,7],[8],[9],[10,11],[12],[13,14],[15,...]\}$, $\mathcal{I}_1 = \{[3,7],[8,9],[10,12],[13,...]\}$, $\mathcal{I}_2 = \{[8,12],[13,...]\}$ and $\mathcal{I}_3 = \{[13,...]\}$. We note that the intervals belonging to the same group are consecutive intervals that do not overlap with each other. Thus, the number of simultaneously running experts is limited by the number of groups, which is bounded by $\log_2 S + 1 = O(\log T)$.

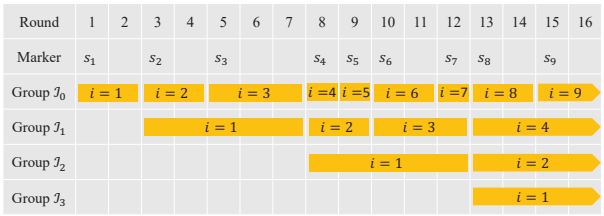


Fig. 4: Illustration of SOGD intervals.

Algorithm 1 Online IS-RRM

Initialize $\mathcal{K}_1 = \{1\}, R_{0,1} = S_{0,1} = 0$.

For $t = 1$ **to** T **do**

1: Set $\mathcal{K}_t = \mathcal{K}_{t-1}$ and update $\mathcal{K}_t$ as in (24) and (25).
2: Each expert $k \in \mathcal{K}_t$ updates its decision $\boldsymbol{w}_t^{(k)}$ as in (18) and its weight $p_{t,k}$ as in (28).
3: Calculate and commit the RB allocation $\boldsymbol{w}_t$ as in (30) and receive a loss function F_t as in (14).
4: Each expert $k \in \mathcal{K}_t$ updates $R_{t,k}$ and $S_{t,k}$ as in (26) and (27), respectively.

end

Formally, in each round t, the active experts of the previous round are first maintained, i.e., $\mathcal{K}_t = \mathcal{K}_{t-1}$. Then, if $t = t_k^s$, add expert k to the set of active experts, i.e.,

$$\mathcal{K}_t = \mathcal{K}_t \cup \{k\}, \forall t_k^s = t. \quad (24)$$

If $t > t_k^e$, remove expert k from the set of active experts, i.e.,

$$\mathcal{K}_t = \mathcal{K}_t \backslash k, \forall t_k^e < t. \quad (25)$$

Finally, we present the meta-algorithm. For any round t, the cumulative regret of the base station with respect to expert $k \in \mathcal{K}_t$ is recursively given by

$$R_{t,k} = R_{t-1,k} + F_t(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_t^{(k)}), \quad (26)$$

and the sum of the absolute value of regret with respect to expert k is recursively given by

$$S_{t,k} = S_{t-1,k} + \left|F_t(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_t^{(k)})\right|. \quad (27)$$

We define $R_{t_k^s-1,k} = 0$ and $S_{t_k^s-1,k} = 0$ for all experts k. The weight of expert k in round t is then given by

$$p_{t,k} = \frac{\Phi(R_{t-1,k}, S_{t-1,k})}{\sum_{k\in\mathcal{K}_t}\Phi(R_{t-1,k}, S_{t-1,k})}, \quad (28)$$

where

$$\Phi(x,y) = \exp\left\{\frac{[x+1]_+^2}{3(y+1)}\right\} - \exp\left\{\frac{[x-1]_+^2}{3(y+1)}\right\}, \quad (29)$$

and $[x]_+ = \max(x, 0)$. Note that $S_{t,k} \geq 0$, and $\Phi(x, y)$ is always properly defined. The outcome RB allocation is then given by weighting the decisions of all active experts, i.e.,

$$\boldsymbol{w}_t = \sum_{k \in \mathcal{K}_t} p_{t,k} \boldsymbol{w}_t^{(k)}. \tag{30}$$

The proposed online IS-RRM scheme is summarized in **Algorithm** 1. In each round t, the set of active experts $\mathcal{K}_t$ is updated, and each active expert $k \in \mathcal{K}_t$ suggests an RB allocation as given in (18). The base station commits $\boldsymbol{w}_t$ as in (30) and receives a loss function F_t as in (14). Then $R_{t,k}$ and $S_{t,k}$ are updated by (26) and (27), respectively. The computational complexity mainly comes from the projection operation as given in (19). Since the constraint $\boldsymbol{w} \in \mathcal{W}$ is a convex set in an N-dimensional space and the objective function $\|\boldsymbol{w} - \boldsymbol{w'}\|$ is a quadratic form with an identity Hessian, it is a convex quadratic programming problem, which can be efficiently solved by interior-point methods in $O(\log N)$ iterations from computational experience [43]. Note that the number of simultaneously active experts is limited by $O(\log T)$. The computational complexity of the proposed online IS-RRM algorithm is then given by $O(\log N \log T)$.

B. Adaptive Regret Analysis

SACS has proven to have a sublinear adaptive regret, which is formally given by the following theorem [42].

Theorem 1. *Under Assumptions 1, 2 and 3, for any interval $[r, s] \subseteq [T]$ and any $\boldsymbol{x} \in \mathcal{X}$, SACS satisfies*

$$\begin{aligned}&\sum_{t=r}^{s} [f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x})] \\ =&O\left(\sqrt{\log \sum_{t=1}^{s} f_t(\boldsymbol{x}) \left(\sum_{t=r}^{s} f_t(\boldsymbol{x})\right) \log \sum_{t=r}^{s} f_t(\boldsymbol{x})}\right).\end{aligned} \tag{31}$$

For the considered IS-RRM problem, the three assumptions are satisfied, and we have the following corollary.

Corollary 1. *The adaptive regret of the proposed online IS-RRM algorithm satisfies*

$$\textit{A-Regret}(T, \tau) = O\left(\sqrt{\log(R_T T)(R_\tau^* \tau) \log(R_\tau^* \tau)}\right), \tag{32}$$

where

$$R_T = \frac{1}{T} \sum\nolimits_{p \in \mathcal{P}_{[1,T]}} B_p, \tag{33}$$

is the average traffic rate in total T rounds and

$$R_\tau^* = \max_{[\tau] \in [T]} \left(\frac{1}{\tau} \sum\nolimits_{p \in \mathcal{P}_{[\tau]}} B_p\right), \tag{34}$$

is the maximum traffic rate in any interval with length τ.

Proof. For the loss function defined in (14), we have

$$\begin{aligned}\sum_{t=r}^{s} F_t(\boldsymbol{w}) &\approx \frac{1}{M} \sum_{t=r}^{s} \sum_{n=1}^{N} \alpha_{t,n} \theta_{t,n}(w_n) \\ &\leq \frac{1}{Mr^*} \sum_{t=r}^{s} \sum_{n=1}^{N} \theta_{t,n}(w_n) \\ &\leq \frac{1}{Mr^*} \sum\nolimits_{p \in \mathcal{P}_{[r,s]} \cup \mathcal{Q}_r} \Theta_{p,s} \\ &\leq \frac{1}{Mr^*} \left(\sum_{p \in \mathcal{P}_{[r,s]}} B_p + \sum_{p \in \mathcal{Q}_r} B_p\right).\end{aligned} \tag{35}$$

Note that $\sum_{p \in \mathcal{Q}_r} B_p$ representing the total amount of buffered bits in round t is always bounded. For any interval $[r, s]$ with length τ, we have

$$\sum_{t=r}^{s} F_t(\boldsymbol{w}) = O\left(\sum_{p \in \mathcal{P}_{[r,s]}} B_p\right) = O\left(R_{[r,s]} \cdot \tau\right), \tag{36}$$

where $R_{[r,s]} = (1/\tau) \sum_{p \in \mathcal{P}_{[r,s]}} B_p$ is the traffic rate during interval $[r, s]$. By substituting (36) into (31), we have

$$\begin{aligned}&\textit{A-Regret}(T, \tau) \\ =&\max_{[\tau] \in [T]} \textit{Regret}([\tau]) \\ =&\max_{[\tau] \in [T]} O\left(\sqrt{\log\left(R_{[1,s]} \cdot s\right)\left(R_{[r,s]} \cdot \tau\right) \log\left(R_{[r,s]} \cdot \tau\right)}\right) \\ =&O\left(\sqrt{\log(R_T T)(R_\tau^* \tau) \log(R_\tau^* \tau)}\right).\end{aligned} \tag{37}$$

□

Corollary 1 indicates that the adaptive regret of the proposed online IS-RRM algorithm is sublinear to both the interval length τ and the total time horizon T. Thus, it is a low-adaptive-regret algorithm. Also, the adaptive regret increases sublinearly with R_T and R_τ^*, which implies that a performance degradation can be expected when the base station suffers from a high traffic load or a large traffic burst.

C. Other Performance Metrics

In this paper, we consider slices with a uniform SLA metric, i.e., the PLR under a strict delay budget. However, a practical network may involve slices with different SLA metrics, e.g., enhanced mobile broadband slices with the average data rate and ultra reliable and low latency communication slices with the average packet delay. There may even exist a slice with multiple SLA metrics, e.g., an automatic driving slice with KPIs in terms of both the data rate and the packet delay. In order to map the heterogeneous SLAs into a uniform loss function, the SLA violation ratio can be used as a general metric, which is always between 0 and 1 due to the statistical wireless environment. The slice-specific loss can be defined as a convex function of the gap between the actual violation ratio and the upper bound 1, and the system loss can be defined as the weighted sum loss of all slices. Therefore, different types of slices can be incorporated in a general OCO framework and be treated differently according to their weights.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3087127, IEEE Transactions on Communications

VI. Simulation Results

In this section, we evaluate the proposed online IS-RRM algorithm and compare it with two benchmark algorithms, i.e., the optimal static algorithm that maintains a static RB allocation that is optimal for total T rounds, i.e.,

$$\boldsymbol{w}^{os} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \sum_{t=1}^{T} F_t(\boldsymbol{w}), \tag{38}$$

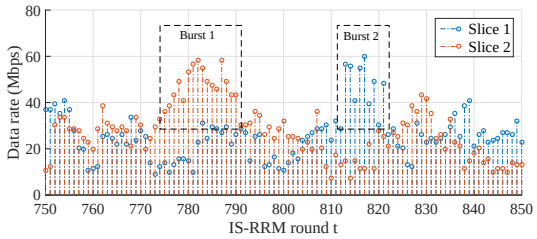
and the optimal dynamic algorithm that performs the optimal RB allocation in each round t, i.e.,

$$\boldsymbol{w}_t^{od} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} F_t(\boldsymbol{w}). \tag{39}$$

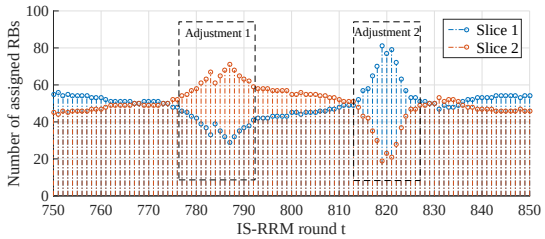
We note that the optimality of the benchmark algorithms can only be achieved in hindsight, and thus, they are not applicable for practical scenarios. Specifically, $\boldsymbol{w}^{os}$ can be seen as the best we can do when only long-term statistical information is available, e.g., the average traffic rate of each slice, while $\boldsymbol{w}_t^{od}$ represents the optimal myopic strategy when the system dynamics in the next round can be perfectly predicted. Here, the truly optimal strategy, which achieves the minimum cumulative loss among all feasible allocation sequences, is not provided, since it requires not only a perfect prediction of the system during total T allocation windows but also exponentially growing computing power for solving the dynamic programming problem behind.

We consider an LTE cell with total 20 MHz bandwidth, where the time-frequency resources are organized into RBs (or more strictly speaking, RB pairs) of 1 ms times 180 kHz. The transmitted signal occupies 90% of the channel bandwidth, and the total number of RBs in each TTI is given by $W = 100$. The number of UEs of each slice is uniformly given by 10. The slice-specific scheduler is uniformly given by a proportional fair scheduler with a fairness window 100 ms and a maximum packet delay $d_n = 100$ ms. The spectrum efficiency of each UE-pRB pair is assumed to be independently and uniformly distributed between 2 bps/Hz and 5 bps/Hz. Without explicit mention, the length of IS-RRM allocation window is set as $L = 10$ TTIs, or equally 10 ms.

The packet size is uniformly given by 1024 bytes. The maximal average throughput that can be supported without packet loss is then given by $\lambda_0 = W \times [(2+5)/2] \times (180000 \times 10^{-3})/(1024 \times 8) = 7.69$ packets per TTI. The packet arrival rate of each UE is uniformly given by $\lambda_{UE} = \lambda_0\rho/(NK)$, where $\rho < 1$ is the traffic load parameter that represents the ratio of the network throughput to the maximal throughput. We assume the traffic of each UE alternatively follows a high data rate pattern and a low data rate pattern, the lengths of which are assumed to be uniformly distributed in $[1000, 2000]$ ms and $[100, 200]$ ms, respectively. Each traffic pattern is represented by a poisson process with a constant packet arrival rate λ^{high} or λ^{low}, and we define $\kappa = \lambda^{high}/\lambda^{low}$ as the data rate ratio between the high and low traffic patterns. Thus, we have $\lambda^{high} = 11\lambda_0\rho\kappa/[NK(\kappa+10)]$ and $\lambda^{low} = 11\lambda_0\rho/[NK(\kappa+10)]$. Note that ρ and κ reflect the load and dynamics of user traffic, which correspond to the average data rate R_T and the burst data rate R_τ^*, respectively.



(a) Data rate in different IS-RRM rounds.



(b) RB allocation in different IS-RRM rounds.

Fig. 5: Illustration of the proposed online IS-RRM algorithm in a two-slice network with $\rho = 0.8, \kappa = 10$ and $r_1 = r_2 = 0.01$.

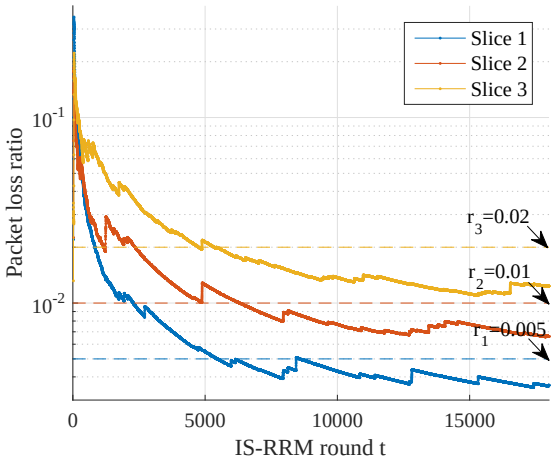
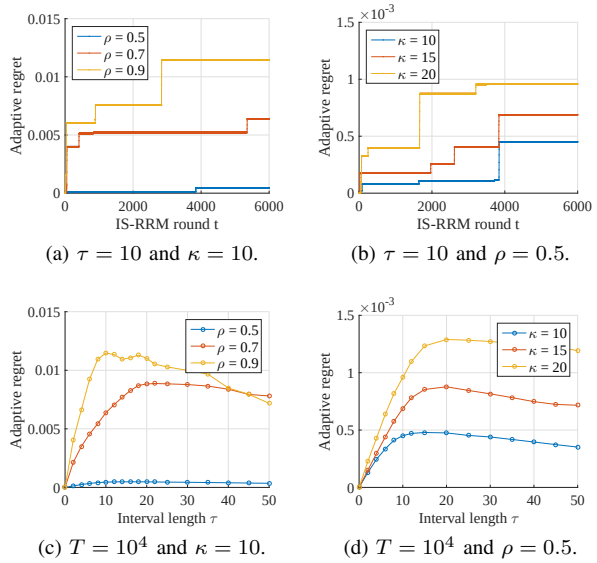


Fig. 6: Packet loss ratio as a function of t with $\rho = 0.9, \kappa = 20, r_1 = 0.005, r_2 = 0.01$ and $r_3 = 0.02$.

A. Illustrative Example

In Fig. 5, we show the dynamic traffic rate and the corresponding RB allocation of the proposed online IS-RRM algorithm in a two-slice network with $\rho = 0.8$ and $\kappa = 10$. The target PLRs are uniformly given by $r_1 = r_2 = 0.01$. As we can see, when traffic bursts arrive in the network (bursts 1 and 2 in Fig. 5a), the proposed algorithm can rapidly adjust the RB allocation by assigning more RBs to the heavily loaded slice (adjustments 1 and 2 in Fig. 5b). When the traffic rate is kept low, as can be seen in other parts of Fig. 5a, each SOGD expert maintains a steady RB allocation and the outcome RB allocation becomes steady, as shown in other parts of Fig. 5b. Thus, Fig. 5 shows that the proposed online IS-RRM algorithm can keep track of the environment changes and adjust the RB allocation accordingly.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3087127, IEEE Transactions on Communications



(a) $\tau = 10$ and $\kappa = 10$. (b) $\tau = 10$ and $\rho = 0.5$.

(c) $T = 10^4$ and $\kappa = 10$. (d) $T = 10^4$ and $\rho = 0.5$.

Fig. 7: Adaptive regret as a function of t and τ with $\rho = 0.5, 0.7, 0.9, \kappa = 10, 15, 20$, and $r_1 = r_2 = 0.01$.

B. Convergence

In Fig. 6, we show the PLR of each slice as a function of t with the proposed online IS-RRM algorithm in a three-slice network with $\rho = 0.9$ and $\kappa = 20$. The target PLRs are given by $r_1 = 0.005, r_2 = 0.01$ and $r_3 = 0.02$, respectively. As we can see, except for those peak points with traffic bursts and inevitable packet losses, the PLRs roughly decrease with t, which implies that the slice performance can be gradually improved as the learner receives more feedback data and adjusts the RB allocation accordingly. Also, we see that all three slices converge to values below their target PLRs after about 10^4 rounds (or equally, 100 s), which implies that the proposed online IS-RRM algorithm can effectively distinguish the diverse requirements of different slices and provide differentiated services accordingly.

C. Adaptive Regret

In Fig. 7, we show the adaptive regret as a function of t and τ with the proposed online IS-RRM algorithm in a two-slice network. The traffic load parameter is set as $\rho = 0.5, 0.7, 0.9$ and the data rate ratio is set as $\kappa = 10, 15, 20$. The target PLRs are uniformly given by $r_1 = r_2 = 0.01$. As we can see, the adaptive regret increases sublinearly with t and τ as the proposed algorithm is a low-adaptive-regret algorithm. In particular, we see that the adaptive regret decreases when τ exceeds a certain threshold, which implies that the performance gap between the proposed algorithm and the optimal static decision decreases as the measurement interval is increased. In fact, the proposed algorithm can outperform the optimal static algorithm, as we will see in the following simulation results. In addition, we see that when ρ and κ are increased, the adaptive regret is also increased, which verifies our analysis in Corollary 1 that the adaptive regret increases with the average traffic rate R_T and the burst traffic rate R_τ^*.

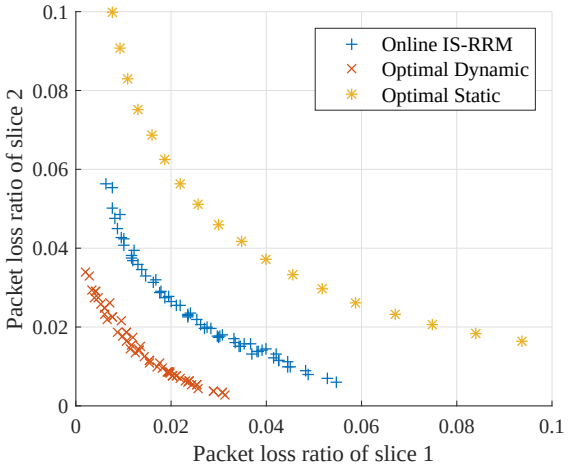


Fig. 8: Pareto optimal front with $\rho = 0.9, \kappa = 20$ and $T = 10^4$.

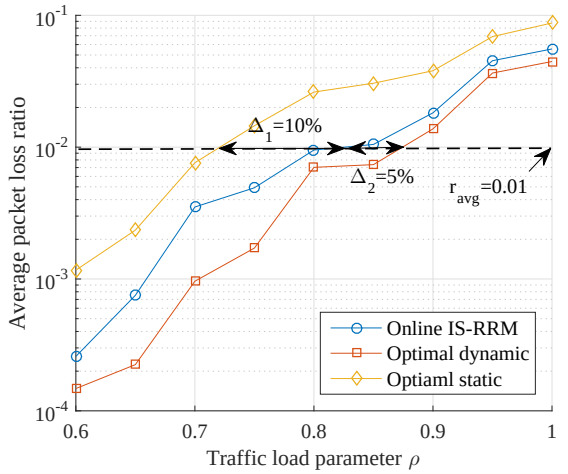


Fig. 9: Average packet loss ratio as a function of traffic load parameter ρ with $\kappa = 20$ and $T = 10^4$.

D. Efficiency

In Fig. 8, we show the Pareto optimal fronts of the proposed online IS-RRM algorithm as well as the benchmark algorithms in a two-slice network with $\rho = 0.9$ and $\kappa = 20$. The Pareto optimal fronts are obtained by randomly choosing each slice's target PLR between 10^{-4} and 10^{-1}. As we can see, since both slices compete for the limited radio resources, the PLR of one slice is always reduced when the PLR of the other slice increases. As the proposed algorithm can adjust according to the time-varying traffic, it outperforms the optimal static algorithm, i.e., both slices can achieve a lower PLR than the best static allocation $\boldsymbol{w}^{os}$. Also, as the instant loss function cannot be obtained in time, the proposed algorithm underperforms the optimal dynamic algorithm, i.e., both slices will suffer a higher PLR than the best myopic allocation sequence $\{\boldsymbol{w}_t^{od}\}_t$.

In Fig. 9, we show the average PLR of all slices as a function of traffic load parameter ρ in a two-slice network

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3087127, IEEE Transactions on Communications

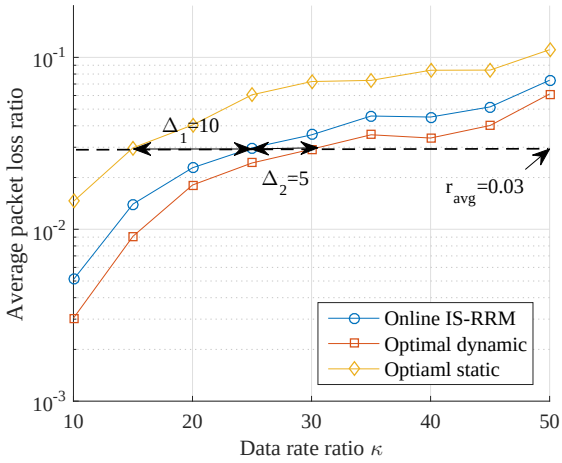


Fig. 10: Average packet loss ratio as a function of data rate ratio κ with $\rho = 0.9$ and $T = 10^4$.

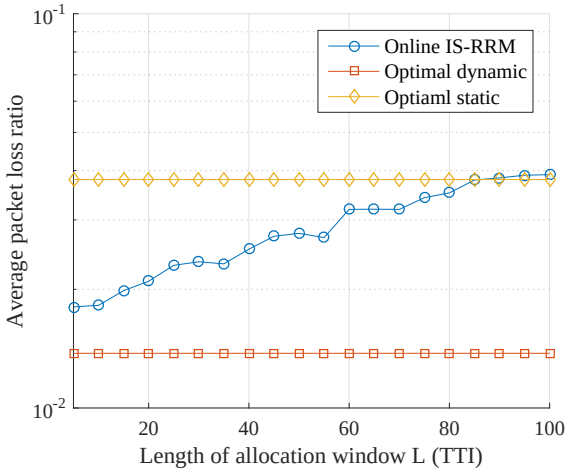


Fig. 11: Average packet loss ratio of all slices as a function of allocation window L with $\rho = 0.9, \kappa = 20$ and $T = 10^4$.

with $\kappa = 20$ and $T = 10^4$. The target PLRs are uniformly given by $r_1 = r_2 = 0.01$. As the total traffic load increases, the average PLR of all slices is always increased due to the intensifying competition for radio resources. To achieve the target PLR 0.01 for all slices, the proposed algorithm can support 82% of the maximal throughput, which is 10% larger than the optimal static algorithm due to the multiplexing gain obtained from the dynamic radio resource allocation and 5% less than the optimal dynamic algorithm due to the lack of information about the instant network behaviors.

In Fig. 10, we show the average PLR of all slices as a function of data rate ratio κ in a two-slice network with $\rho = 0.9$ and $T = 10^4$. The target PLRs are uniformly given by $r_1 = r_2 = 0.03$. As κ increases, the network experiences a higher level of traffic fluctuation, and the average PLR of all slices is increased due to the increasing number of peak periods. If the peak period lasts for a certain number of allocation windows, the proposed algorithm can detect the traffic changes from historical loss functions and adjust the allocation accordingly to reduce packet losses. If the peak period lasts only for a short time, the proposed algorithm cannot predict its appearance from historical data, and the network suffers from a large number of packet losses. To achieve the target PLR 0.03 for all slices, the proposed algorithm allows the high pattern data rate to be 25 times as large as the low pattern data rate, which outperforms the optimal static algorithm by 5 and underperforms the optimal dynamic algorithm by 10.

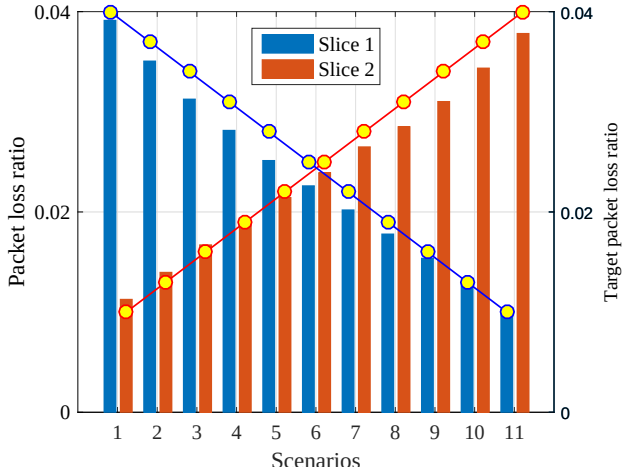


Fig. 12: Packet loss ratio in different scenarios with $\rho = 0.9, \kappa = 20$ and $T = 10^4$.

In Fig. 11, we show the average PLR of all slices as a function of the allocation window L in a two-slice network with $\rho = 0.9, \kappa = 20$ and $T = 10^4$. The target PLRs are uniformly given by $r_1 = r_2 = 0.03$. As we can see, as the length of the allocation window increases, the computational complexity of the proposed algorithm is decreased, and the corresponding performance is degraded. When $L = 10$ ms, the proposed algorithm can closely follow traffic fluctuations and make timely allocation adjustments, and thus its performance is close to the optimal dynamic algorithm using hindsight data. When $L = 100$ ms, the proposed algorithm needs to maintain a static allocation for a large allocation window, which makes its performance close to the optimal static algorithm that always maintains the best static allocation. Therefore, in practical networks, the allocation window should be decided by considering the tradeoff between computational complexity and network performance.

E. Robustness

In Fig. 12, we show the PLR of each slice in a two-slice network with $\rho = 0.9, \kappa = 20$ and $T = 10^4$. From scenario 1 to scenario 11, the target PLR of slice 1 is decreased from $r_1 = 0.04$ to $r_1 = 0.01$, while the target PLR of slice 2 is increased symmetrically from $r_2 = 0.01$ to $r_2 = 0.04$. As we can see, the outcome PLRs are aligned with the target PLRs for both slices. Thus, the proposed online IS-RRM algorithm can achieve a robust performance as the slice requirements vary with time in practical networks. We note that the robustness also depends on the success of multi-slice access control, which ensures

that the network traffic can always be supported with satisfied SLAs. Here, we assume such access control exists while the implementation details are beyond the scope of the paper.

VII. Conclusion

In this paper, we have proposed a novel OCO framework for IS-RRM, where an online IS-RRM scheme is proposed based on the state-of-the-art online learning algorithm. The proposed OCO scheme can gradually learn the environmental changes from the experience data without using sophisticated statistical models and achieve a bounded performance by exploiting the derivatives of well-designed loss functions. Both theoretical analysis and simulation results show that the proposed online IS-RRM algorithm can achieve comparable performance to the optimal strategies in hindsight with various levels of traffic loads and dynamics. Also, the outcome slice performance is shown to be robust to the time-varying SLA requirements as in practical deployments.

Acknowledgement

The authors would like to thank the editors and the anonymous reviewers, whose invaluable comments helped improve the presentation of this paper substantially.

References

[1] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. M. Leung, "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[2] S. Zhang, "An Overview of Network Slicing for 5G," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 111–117, Jun. 2019.

[3] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[4] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN Slicing for Verticals: Enablers and Challenges," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 28–34, Jan. 2019.

[5] 3GPP TS 38.300, "NR and NG-RAN Overall Description; (Release 15)," Mar. 2018.

[6] A. Ksentini and N. Nikaein, "Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 102–108, Jun. 2017.

[7] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On Radio Access Network Slicing from a Radio Resource Management Perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, Oct. 2017.

[8] S. D'Oro, F. Restuccia, and T. Melodia, "Toward Operator-to-Waveform 5G Radio Access Network Slicing," *IEEE Commun. Mag.*, vol. 58, no. 4, pp. 18–23, Apr. 2020.

[9] C. Chang and N. Nikaein, "RAN Runtime Slicing System for Flexible and Dynamic Service Execution Environment," *IEEE Access*, vol. 6, pp. 34 018–34 042, Jun. 2018.

[10] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.

[11] T. Guo and R. Arnott, "Active LTE RAN Sharing with Partial Resource Reservation," in *Proc. IEEE VTC'13*, Las Vegas, NV, USA, Sept. 2013.

[12] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization," in *Proc. IEEE INFOCOM'17*, Atlanta, GA, USA, May 2017.

[13] T. Guo and A. Surez, "Enabling 5G RAN Slicing With EDF Slice Scheduling," *IEEE Trans. Veh. Tech.*, vol. 68, no. 3, pp. 2865–2877, Mar. 2019.

[14] M. I. Kamel, L. B. Le, and A. Girard, "LTE Wireless Network Virtualization: Dynamic Slicing via Flexible Scheduling," in *Proc. IEEE VTC'14*, Vancouver, Canada, Sept. 2014.

[15] A. Lieto, I. Malanchini, A. Walid, and A. Capone, "Quantifying the Gain of Dynamic Network Slicing under Stringent Constraints," in *Proc. IEEE GLOBECOM'19*, Waikoloa, HI, USA, Dec. 2019.

[16] J. Tang, B. Shim, and T. Q. S. Quek, "Service Multiplexing and Revenue Maximization in Sliced C-RAN Incorporated With URLLC and Multicast eMBB," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 881–895, Apr. 2019.

[17] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Prez, "Resource Sharing Efficiency in Network Slicing," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 3, pp. 909–923, Sept. 2019.

[18] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The Slice is Served: Enforcing Radio Access Network Slicing in Virtualized 5G Systems," in *Proc. INFOCOM'19*, Paris, France, Apr. 2019.

[19] M. Zambianco and G. Verticale, "Interference Minimization in 5G Physical-Layer Network Slicing," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4554–4564, Jul. 2020.

[20] X. Foukas, M. K. Marina, and K. Kontovasilis, "Iris: Deep Reinforcement Learning Driven Shared Spectrum Access Architecture for Indoor Neutral-Host Small Cells," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1820–1837, Aug. 2019.

[21] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep Reinforcement Learning With Discrete Normalized Advantage Functions for Resource Management in Network Slicing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1337–1341, Aug. 2019.

[22] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[23] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The LSTM-Based Advantage Actor-Critic Learning for Resource Management in Network Slicing With User Mobility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sept. 2020.

[24] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Prez, "Multi-Tenant Radio Access Network Slicing: Statistical Multiplexing of Spatial Loads," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 3044–3058, Oct. 2017.

[25] J. Zheng, P. Caballero, G. de Veciana, S. J. Baek, and A. Banchs, "Statistical Multiplexing and Traffic Shaping Games for Network Slicing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2528–2541, Dec. 2018.

[26] P. Caballero, A. Banchs, G. de Veciana, X. Costa-Prez, and A. Azcorra, "Network Slicing for Guaranteed Rate Services: Admission Control and Resource Allocation Games," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6419–6432, Oct. 2018.

[27] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Prez, "Network Slicing Games: Enabling Customization in Multi-Tenant Mobile Networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 662–675, Apr. 2019.

[28] E. Hazan, "Introduction to Online Convex Optimization," *Found. Trends Opt.*, vol. 2, no. 3-4, pp. 157–325, Aug. 2016.

[29] E. C. Hall and R. M. Willett, "Online Convex Optimization in Dynamic Environments," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, Jun. 2015.

[30] T. Chen, Q. Ling, and G. B. Giannakis, "An Online Convex Optimization Approach to Proactive Network Resource Allocation," *IEEE Trans. Signal Process.*, vol. 65, no. 24, pp. 6350–6364, Dec. 2017.

[31] A. S. Bedi, P. Sarma, and K. Rajawat, "Tracking Moving Agents via Inexact Online Gradient Descent Algorithm," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 202–217, Feb. 2018.

[32] X. Cao, J. Zhang, and H. V. Poor, "A Virtual-Queue-Based Algorithm for Constrained Online Convex Optimization With Applications to Data Center Resource Allocation," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 703–716, Aug. 2018.

[33] B. Li, T. Chen, and G. B. Giannakis, "Secure Mobile Edge Computing in IoT via Collaborative Online Learning," *IEEE Trans. Signal Process.*, vol. 67, no. 23, pp. 5922–5935, Dec. 2019.

[34] M. Zhou, T. Wang, and S. Wang, "Spectrum Sensing Across Multiple Service Providers: A Discounted Thompson Sampling Method," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2402–2406, Dec. 2019.

[35] Y. Lin, T. Wang, and S. Wang, "UAV-Assisted Emergency Communications: An Extended Multi-Armed Bandit Perspective," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 938–941, May 2019.

[36] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed Online Convex Optimization With Time-Varying Coupled Inequality Constraints," *IEEE Trans. Signal Process.*, vol. 68, pp. 731–746, Jan. 2020.

[37] Z. Sun and M. R. Nakhai, "An Online Learning Algorithm for Distributed Task Offloading in Multi-Access Edge Computing," *IEEE Trans. Signal Process.*, vol. 68, pp. 3090–3102, Apr. 2020.

[38] Z. Kuai and S. Wang, "Thompson Sampling-Based Antenna Selection With Partial CSI for TDD Massive MIMO Systems," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7533–7546, Dec. 2020.
[39] A. Ksentini, P. A. Frangoudis, A. PC, and N. Nikaein, "Providing Low Latency Guarantees for Slicing-Ready 5G Systems via Two-Level MAC Scheduling," *IEEE Netw.*, vol. 32, no. 6, pp. 116–123, Nov./Dec. 2018.
[40] R. T. Marler and S. A. Jasbir, "Survey of Multi-Objective Optimization Methods for Engineering," *Struct. Multidisc. Optim.*, vol. 26, no. 6, pp. 369–395, Mar. 2004.
[41] A. Daniely, A. Gonen, and S. Shalev-Shwartz, "Strongly adaptive online learning," in *Proc. ICML'15*, Lille, France, Jul. 2015.
[42] L. Zhang, T.-Y. Liu, and Z.-H. Zhou, "Adaptive Regret of Convex and Smooth Functions," in *Proc. ICML'19*, Los Angeles, United States, Jun. 2019.
[43] J. Gondzio, "Interior Point Methods 25 Years Later," *Europ. J. Oper. Res.*, vol. 218, no. 3, pp. 587–601, May 2012.

Tianyu Wang (S'11-M'16) received the PhD degree from Peking University, Beijing, China, in 2011. He is currently an associate researcher with the School of Electronic Science and Engineering, Nanjing University, China. He has published more than 40 IEEE journal and conference papers, and received the Best Paper Award from the IEEE ICC'15, IEEE GLOBECOM'14, and ICST ChinaCom'12. His current research interest includes network slicing and machine learning in wireless networks.

Shaowei Wang (S'06-M'07-SM'13) received the PhD degree from Wuhan University, Wuhan, China, in 2006, and joined the School of Electronic Science and Engineering at Nanjing University, Nanjing, China, as a faculty member in the same year, where he is currently a Full Professor. From 2012 to 2013, he was a Visiting Scholar/Professor with Stanford University, Stanford, CA, USA, and The University of British Columbia, Vancouver, BC, Canada. His research interests include communications and networking, operations research and machine learning.