Inter-Slice Radio Resource Management via Online Convex Optimization

Tianyu Wang, Wentao Yu and Shaowei Wang
School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
Email: tianyu.alex.wang@nju.edu.cn, wentaoyu@smail.nju.edu.cn, wangsw@nju.edu.cn

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500493

***Abstract*—Radio access network (RAN) slicing is one of the key technologies in 5G and beyond mobile networks, where multiple logical RANs, also referred to as 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 cells and the diverse requirements of RAN slices, inter-slice radio resource management (IS-RRM) is a highly challenging task. In this paper, we propose a novel online convex optimization (OCO) framework for the IS-RRM, where the instant resource allocation is learned by using historical data revealed from previous allocations. Compared with the existing methods, OCO is an online optimization process that can avoid sophisticated modeling and tuning in highly complicated and dynamic environments. Specifically, a low-complexity online IS-RRM algorithm is proposed, which employs multiple expert-algorithms running parallelly to keep track of environmental changes. Simulation results show that the proposed method can provide efficient IS-RRM with a comparable performance to the optimal strategies in hindsight.**

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

I. Introduction

Multi-slice radio resource management (RRM) is considered to be critical for the success of radio access network (RAN) slicing [1]. Due to the diverse requirements of RAN slices, the conventional medium access control (MAC) scheduler, which directly distributes radio resources to each individual user in each transmission time interval (TTI), becomes a multi-objective optimization problem that is too complex to solve [2]. Instead, multi-slice RRM is decomposed into inter-slice and intra-slice RRM, which can be performed separately across different time scales. Intra-slice RRM, which is responsible for allocating slice-dedicated resources to the corresponding slice users in each TTI, focuses on the packet-level QoS performance of each user and can be realized by using conventional MAC schedulers. Inter-slice RRM, which is abbreviated as IS-RRM in this paper, is responsible for allocating radio resources to each slice in each allocation window. IS-RRM focuses on the satisfaction of the service level agreement (SLA) of each slice and has become a new technical challenge.

Network virtualization substrate is one of the earliest work in slice-based RRM, which operates at MAC-frame granularity to decouple the slice scheduling problem from the packet scheduling problem [3]. However, the per-slice utility function is defined based on the average resource usage, which is not suitable for RANs with highly dynamic user traffic and network environments. To achieve more efficient RAN slicing with finer granularity, mobile traffic forecasting technology is introduced to predict the actual footprint of each particular slice [4], and short-term KPIs are considered by transforming the SLAs into the requirements of physical radio resources in each allocation period [5]. Also, the large-scale resource sharing efficiency is studied by extending the allocation window into hours, where the SLA requirements are uniformly described by using a guaranteed demand parameter and an overbooking penalty parameter [6]. Due to the complexity of RAN slicing, these model-based optimization methods usually generate a highly complicated optimization problem that is difficult to solve, where simplification techniques and heuristic algorithms are widely used in the literature [4–8].

Machine learning has attracted a lot of attention in the study of 5G and beyond networks [9, 10]. To avoid the drawbacks of model-based IS-RRM, data-driven methods based on deep reinforcement learning are recently proposed [11, 12]. Such methods treat IS-RRM as a Markov decision process and utilize deep reinforcement learning algorithms to decide the best allocation strategy. In [11], a deep reinforcement learning-based dynamic pricing mechanism is proposed to provide incentives for the slice owners to efficiently share the spectrum resources. In [12], deep reinforcement learning-based bandwidth allocation algorithms are proposed, where discrete normalized advantage functions and generative adversarial networks are introduced to accelerate the convergence rate and improve the approximation accuracy, respectively. However, due to the complexity of wireless networks, the deep neural network needs to be periodically tuned with online data, such that the difference between the offline simulation and the realtime network can be captured. During these tuning periods, these learning algorithms need to extensively explore the action space, which may cause severe SLA violations.

In this article, we consider IS-RRM and develop a novel mathematical framework for the dynamic assignment of downlink bandwidth resources based on online convex optimization (OCO) technique [13–15]. Compared with the classic opti-

This work was partially supported by the National Natural Science Foundation of China (61801208, U1936202, 61931023).

mization methods using gradient descend techniques, OCO formulates IS-RRM as an online process that can gradually learn the network dynamics from the data of previous allocations, which avoids the modeling difficulty and achieves low computational complexity. Compared with the reinforcement learning methods, OCO averts the massive exploration in action space by exploiting the derivatives of well-defined loss functions, and provides a theoretical performance guarantee in terms of "regret". Specifically, we propose an online IS-RRM algorithm based on the state-of-the-art strongly adaptive algorithm for convex and smooth functions [16]. Simulation results show that the proposed scheme can achieve a comparable performance to the optimal strategies in hindsight.

II. System Model

We consider the downlink scenario of a single cell with total N network slices, where each slice provides a customized network service for K user equipments (UEs). The bandwidth resources are organized into W resource blocks (RBs) in each TTI, and the length of an allocation window is L TTIs. We denote by $\boldsymbol{w}_t = (w_{t,1}, \ldots, w_{t,N})$ the RB allocation of allocation window t, or round t, where $w_{t,n}$ is the number of dedicated RBs of slice n. For any $t \in [1, T]$, we have $\boldsymbol{w}_t \in \mathcal{W}_{\mathbb{Z}}$, where

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

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 period, by $\mathcal{P}_{t,n,k}$ the set of packets that arrive at the traffic queue during round t, and by $\mathcal{Q}_{t,n,k}$ the set of packets that are buffered in the traffic queue at the beginning of round t. For simplicity, we denote $\mathcal{H}_{t,n} = \{\mathcal{H}_{t,n,k}\}_k$, $\mathcal{P}_{t,n} = \{\mathcal{P}_{t,n,k}\}_k$ and $\mathcal{Q}_{t,n} = \{\mathcal{Q}_{t,n,k}\}_k$.

In each TTI, a two-level MAC scheduler is applied [2], which is composed of N slice-specific schedulers as well as a common scheduler. Each slice-specific scheduler is responsible for provisioning packet-level QoS performance, 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 algorithm of slice 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, which can maximize the inter-slice multi-user gain. Note that the size of vRB is aligned with the physical RB (pRB) and the number of pRBs dedicated to slice n in round t is limited by $w_{t,n}$. Compared with the conventional schedulers that jointly schedule all packets in one pass, the two-level scheduler utilizes multiple slice-specific schedulers in parallel, which decouples the multi-dimensional scheduling problem into multiple single-dimensional scheduling problems, and achieves both high flexibility and low computational complexity.

The packet loss ratio (PLR) is considered as the KPI of the SLA between the network operator and the slice tenants. The PLR is defined as the ratio of the number of lost packets to the number of total arriving packets. For any slice n, the number of packets arriving at round t is given by $|\mathcal{P}_{t,n}|$. The number of lost packets in round t, denoted by $C_{t,n}$, depends on traffic state $\mathcal{P}_{t,n}$ and $\mathcal{Q}_{t,n}$, channel state $\mathcal{H}_{t,n}$, bandwidth resources $w_{t,n}$, as well as the slice-specific scheduling algorithm $\mathcal{V}_n$, i.e., $C_{t,n}(\mathcal{P}_{t,n}, \mathcal{Q}_{t,n}, \mathcal{H}_{t,n}, w_{t,n}|\mathcal{V}_n)$. Thus, the PLR of slice n after round t is given by

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

Therefore, IS-RRM can be formulated as a multi-objective programming problem that aims to minimize the PLR of all slices, which is formally written as

$$\min_{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T} (r_{T,1}, r_{T,2}, \ldots, r_{T,N}) \quad (3a)$$

$$\text{s.t.} \quad \boldsymbol{w}_t \in \mathcal{W}_{\mathbb{Z}}, \ t = 1, 2, \ldots, T. \quad (3b)$$

Typically, there does not exist a feasible allocation sequence that simultaneously minimizes $r_{T,n}$ for all $n \in \{1, 2, \ldots, N\}$. Instead, Pareto optimal solutions are usually considered, which are defined as feasible solutions that cannot be improved in any of the objectives $r_{T,n}$ without degrading at least one of the other objectives $r_{T,n'}, n' \neq n$. We note that there are two major drawbacks of the above formulation. The first drawback is that, considering the complexity of slice-specific scheduling algorithms (i.e., $\mathcal{V}_n$), there may not exist an analytical function to formulate the mathematical relationship between the objective $r_{T,n}$ and the optimization variables $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T$, which makes this formulation extremely difficult to solve. The second drawback is that, considering the small granularity of IS-RRM, the network parameters (i.e., $\mathcal{P}_{t,n}, \mathcal{Q}_{t,n}, \mathcal{H}_{t,n}$) can be highly dynamic and hard to predict with statistical models, which may cause severe performance deviation of optimal solutions. In the next section, we introduce the OCO framework to reformulate the IS-RRM problem, which avoids the drawbacks of multi-objective programming.

III. Online IS-RRM Via Online Convex Optimization

We consider the IS-RRM as an online learning process, where the instant RB allocation is learned online by using the data information revealed from previous rounds, rather than being calculated or optimized with comprehensive models that are given beforehand. This online framework allows the base station to learn from the experience and provide better RB allocations as more cases are observed. Formally, we introduce the OCO framework as follows.

A. IS-RRM As OCO

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 design an OCO algorithm that minimizes 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 the *regret*. Formally, for any horizon T, the regret of an OCO algorithm $\mathcal{A}$ is defined as [13]

$$\textit{Regret}_{\mathcal{A}}(T) = \sup_{\{f_t \in \mathcal{F}\}_t} \left\{ \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x}) \right\}. \quad (4)$$

Thus, the regret is defined as the loss difference in the worst case for all possible loss functions.

For the IS-RRM problem, the base station can be seen as a learner that iteratively determines the RB allocation $\boldsymbol{w}_t$ in each round t to optimize the SLA performance of all slices within total T rounds. To use the OCO framework, we relax the feasible set of RB allocations to the convex hull of $\mathcal{W}_{\mathbb{Z}}$, which is a standard N-simplex scaled by W, given by

$$\mathcal{W} = \left\{ \boldsymbol{x} \in \mathbb{R}^N \mid \sum_{n=1}^{N} x_n \leq W \text{ and } x_n \geq 0, \forall n \right\}. \quad (5)$$

Thus, we have $\boldsymbol{w}_t \in \mathcal{W}$ for all $t \in \{1, 2, \ldots, T\}$. Note that constraint (3b) can still be satisfied by rounding the coordinates of $\boldsymbol{w}_t$ before it is committed in round t.

The loss function in each round can be designed based on the PLR performance of each packet in the corresponding TTIs. 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 given by B_p. For simplicity, we denote the l-th TTI of round t by a tuple (t, l). For any TTI (t, l), and the current lifetime of packet p is defined as the time interval between (t_p^{in}, l_p^{in}) and (t, l), denoted by $\mathcal{T}_p(t, l)$. We denote by $B_{p,t,l}$ the amount of p's data bits that are scheduled by the MAC scheduler at TTI (t, l), and have $B_{p,t,l} \geq 0$ and $\sum_{(t,l)} B_{p,t,l} \leq B_p$. For any round $t \in [t_p^{in}, t_p^{out}]$, the cumulative loss of packet p is defined as

$$\Theta_{p,t} = \sum_{(s,m) \in \mathcal{T}_p(t,L)} B_{p,s,m} \left(\frac{d_{p,s,m}}{d_{n_p}} \right)^2 + \left(B_p - \sum_{(s,m) \in \mathcal{T}_p(t,L)} B_{p,s,m} \right) \left(\frac{d_{p,t,L}}{d_{n_p}} \right)^2, \quad (6)$$

where $d_{p,s,m} = |\mathcal{T}_p(s, m)|$ represents the experienced delay of packet p at the end of TTI (s, m) and d_{n_p} represents the packet delay budget of slice n_p, respectively.

As seen in (6), the cumulative loss of packet p consists of two parts, indicating the loss of transmitted and buffered bits, respectively. For each part, the loss value is defined to be proportional to the square of experienced delay normalized by the delay budget, and scaled by the amount of transmitted or buffered bits in the corresponding TTI. Thus, the cumulative loss $\Theta_{p,t}$ can reflect the packet-level PLR performance by indicating how close packet p is about to be lost in the upper layer after the transmission of round t. Specifically, we have $\Theta_{p,t} \in (0, B_p]$ and it is a strictly increasing function of t. For simplicity, we set $\Theta_{p,t_p^{in}-1} = 0$ for all packets. Given the per-packet cumulative loss defined in (6), the loss of slice n in round t is defined as

$$\theta_{t,n} = \sum_{p \in \mathcal{P}_{t,n} \cup \mathcal{Q}_{t,n}} \left(\Theta_{p,t} - \Theta_{p,t-1} \right), \quad (7)$$

which represents total marginal increase of $\Theta_{p,t}$ for all packets of slice n in round t. Denoting $B_{n,t} = \sum_{p \in \mathcal{P}_{t,n}} B_p \cup \sum_{p \in \mathcal{Q}_{t,n}} B_p$ as the data bits involved in the scheduling process of slice n in round t, we have $\theta_{t,n} \in (0, B_{n,t}]$.

For any given number of RBs $w_{t,n}$ assigned to slice n in round t, the loss value $\theta_{t,n}$ can be obtained in hindsight, based on the observed network parameters $\mathcal{P}_{t,n}$, $\mathcal{Q}_{t,n}$ and $\mathcal{H}_{t,n}$, and the slice-specific scheduling algorithm $\mathcal{V}_n$. Intuitively, $\theta_{t,n}$ is a positive value that decreases with $w_{t,n}$, and the gradient approaches to 0 as $w_{t,n}$ goes to infinity. In Fig. 1, we illustrate $\theta_{t,n}$ as a function of $w_{t,n}$ in typical network settings. We can see that a negative exponential function can be well fitted to approximate $\theta_{t,n}$, given by

$$\tilde{\theta}_{t,n}(w_{t,n}) = a_{t,n} \cdot \exp\left\{ b_{t,n} \cdot w_{t,n} \right\}, \quad (8)$$

where $a_{t,n} > 0$ and $b_{t,n} < 0$ are the coefficients of the fitted curve. It is easy to prove that $\tilde{\theta}_{t,n}$ is a strictly decreasing and convex function of $w_{t,n}$, and $0 < \tilde{\theta}_{t,n} \leq a_{t,n}$. Note that the convexity lies in the fact that the slice performance can always be improved by additional bandwidth resources while the marginal improvement always decreases. Thus, the loss function defined in (8) can be extended to other KPIs. In practical deployments, the calculation of curve fitting can be

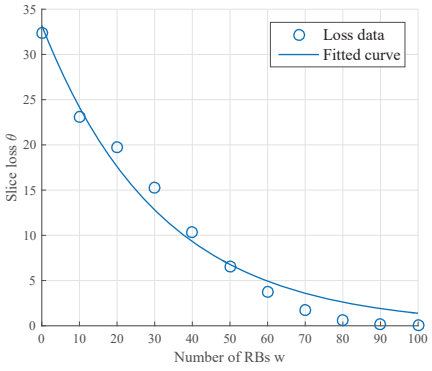

Fig. 1: Illustration of slice loss θ as a function of RB number w. The fitted curve is given by $\theta = 33.21 \times \exp\{-0.03176w\}$ and the coefficient of determination is 0.9773.

simplified by using a lookup table, an experience function or a deep neural network that is well trained offline.

To formulate a single loss function that can reflect the PLR performance of all slices, we introduce a priori weight $\alpha_{t,n}$ for each slice n in each round t. Specifically, we define the weight factor as

$$\alpha_{t,n} = \frac{r_{t-1,n}}{r_n}, \tag{9}$$

where $r_{t-1,n}$ is the PLR of slice n after the previous round $t-1$ and r_n is the target PLR. Therefore, the total loss in round t is defined as

$$F_t(\boldsymbol{w}_t) = \sum_{n=1}^{N} \alpha_{t,n} \tilde{\theta}_{t,n}(w_{t,n}). \tag{10}$$

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 $F_t(\boldsymbol{w}_t)$. In the next sbusection, a low-complexity online IS-RRM algorithm is proposed.

B. Online IS-RRM Algorithm

The proposed online IS-RRM algorithm contains three parts [16]: 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, it can be seen as an ensemble learning method that runs multiple expert-algorithms parallelly in different time intervals, and dynamically combines the decisions of active experts by updating their weights in each round.

We first introduce the expert-algorithm, referred to as scale-free online gradient descent (SOGD). 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 starts at round t_k^s and ends at round t_k^e. The RB allocation of expert k, denoted by $\boldsymbol{w}_{t,k}$, is updated by

$$\boldsymbol{w}_{t,k} = \Pi_{\mathcal{W}} \left[\boldsymbol{w}_{t-1,k} - \eta_{t-1,k} \nabla F_{t-1}(\boldsymbol{w}_{t-1,k}) \right], \tag{11}$$

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.,

$$\Pi_{\mathcal{W}}(\boldsymbol{w'}) \coloneqq \arg\min_{\boldsymbol{w} \in \mathcal{W}} \|\boldsymbol{w} - \boldsymbol{w'}\|, \tag{12}$$

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

$$\eta_{t-1,k} = \frac{D/\sqrt{2}}{\sqrt{\delta + \sum_{s=t_k^s}^{t-1} \|\nabla F_s(\boldsymbol{w}_{s,k})\|^2}}, \tag{13}$$

where $D = \sqrt{2}W$ is the domain diameter of $\mathcal{W}$ and $\delta > 0$ 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.

Algorithm 1 Online IS-RRM

Initialize $\mathcal{K}_1 = \{1\}, R_{0,1} = S_{0,1} = 0$
For $t = 1$ **to** T **do**
  Set $\mathcal{K}_t = \mathcal{K}_{t-1}$ and update $\mathcal{K}_t$ as in (17) and (18)
  Each expert $k \in \mathcal{K}_t$ calculates its RB allocation $\boldsymbol{w}_{t,k}$ as in (11) and its weight $p_{t,k}$ as in (21)
  Calculate and commit the RB allocation $\boldsymbol{w}_t$ as in (23) and receive the loss function F_t as in (10)
  Each expert $k \in \mathcal{K}_t$ updates $R_{t,k}$ and $S_{t,k}$ as in (19) and (20), respectively.
end

The intervals of SOGD experts 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

$$C = 20HD^2 + 2D\sqrt{2\delta}, \tag{14}$$

where parameter H is chosen to ensure all loss functions $\{F_t\}$ are H-smooth. 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 restarts 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

$$\mathcal{I} = \bigcup_{k \in \mathbb{N} \cup \{0\}} \mathcal{I}_k, \tag{15}$$

where for all $k \in \mathbb{N} \cup \{0\}$

$$\mathcal{I}_k = \left\{ \left[s_{i \cdot 2^k}, s_{(i+1) \cdot 2^k} - 1 \right], i \in \mathbb{N} \right\}. \tag{16}$$

We note that the intervals belonging to the same $\mathcal{I}_k$ are consecutive intervals that do not overlap with each other. Thus, the number of simultaneously running experts is given by the number of $\mathcal{I}_k$, which is bounded by $\log_2 S + 1 = O(\log T)$.

Formally, the algorithm inherits the active experts of the previous round, i.e., $\mathcal{K}_t = \mathcal{K}_{t-1}$, add expert k if $t = t_k^s$, i.e.,

$$\mathcal{K}_t = \mathcal{K}_t \cup \{k\}, \forall t_k^s = t. \tag{17}$$

and remove expert k if $t > t_k^e$, i.e.,

$$\mathcal{K}_t = \mathcal{K}_t \backslash k, \forall t_k^e < t. \tag{18}$$

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

$$R_{t,k} = R_{t-1,k} + F_t(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_{t,k}), \tag{19}$$

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

$$S_{t,k} = S_{t-1,k} + |F_t(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_{t,k})|. \tag{20}$$

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

$$p_{t,k} = \frac{\Phi(R_{t-1,k}, S_{t-1,k})}{\sum_{k\in\mathcal{K}_t} \Phi(R_{t-1,k}, S_{t-1,k})}, \tag{21}$$

where

$$\Phi(x,y) = \exp\left\{\frac{[x+1]_+^2}{3(y+1)}\right\} - \exp\left\{\frac{[x-1]_+^2}{3(y+1)}\right\}, \tag{22}$$

and $[x]_+ = \max(x, 0)$. The outcome RB allocation is then given by weighting the decisions of active experts, i.e.,

$$\boldsymbol{w}_t = \sum_{k\in\mathcal{K}_t} p_{t,k}\boldsymbol{w}_{t,k}. \tag{23}$$

The learning process is repeated in each round, and the complete online IS-RRM scheme is shown in **Algorithm** 1.

The computational complexity of the proposed algorithm mainly comes from the projection operation as given in (12). Since the constraint $\boldsymbol{w} \in \mathcal{W}$ is a convex set in N-dimensional space and the objective function $\|\boldsymbol{w} - \boldsymbol{w'}\|$ is a quadratic form with an identity Hessian, the projection is a convex quadratic programming problem, which can be efficiently solved by interior point methods in $O(\log N)$ iterations from computational experience [17]. Note that the number of simultaneously active experts is limited by $O(\log T)$. The computational complexity is then given by $O(\log N \log T)$.

IV. Simulation Results

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 allocation that is optimal for the entire T rounds,

$$\boldsymbol{w}_t^{os} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \sum_{t=1}^{T} F_t(\boldsymbol{w}), \tag{24}$$

and the optimal dynamic algorithm that dynamically performs the optimal allocation in each round t,

$$\boldsymbol{w}_t^{od} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} F_t(\boldsymbol{w}). \tag{25}$$

Note that the benchmark algorithms are not applicable since their optimality can only be achieved in hindsight.

Consider an LTE cell with total $W = 100$ RBs of 1 ms times 180 kHz. The length of an allocation window is given by $L = 10$ TTIs and the number of UEs per slice is given by $K = 10$. All slice-specific schedulers are assumed to be a proportional fair scheduler with fairness window 100 ms and packet delay budget $d_n = 100$ ms. The spectrum efficiency of any UE over any RB is assumed to be independently and uniformly distributed between 2 bps/Hz and 5 bps/Hz. The packet size is fixed at 1024 bytes. The full load cell throughput is 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 in slice n is uniformly given by $\lambda = \lambda_0\rho/(NK)$, where $\rho \leq 1$ is the cell load parameter representing the ratio of the total traffic rate to the full load throughput. The traffic of each UE alternatively follows a low data rate pattern and a high 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^{low} or λ^{high}, and we define $\kappa = \lambda^{high}/\lambda^{low}$ as the rate ratio between high and low traffic patterns. Thus, we have $\lambda^{low} = 11\lambda_0\rho/[NK(\kappa+10)]$ and $\lambda^{high} = 11\lambda_0\rho\kappa/[NK(\kappa+10)]$.

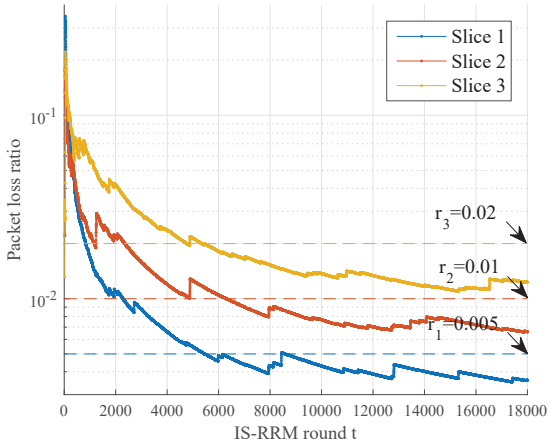

Fig. 3: Packet loss ratio as a function of t with $\rho = 0.9$ and $\kappa = 20$.

In Fig. 3, we show the slice-specific PLR 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. We see that, except for those peak points with packets losses caused by burst traffic, the PLRs roughly decrease with t. Since more feedback information is available as the running time increases, the proposed algorithm can adjust the online RB allocation accordingly and improve the PLR performance gradually. Also, we see that all three slices converge to values below their target PLRs after about 10^4 rounds (or equally, 100 s) and then keep steady, which means all slices receive differentiated but fair treatment in terms of their SLAs. The reason is that the loss function is well-designed such that the minimal value is achieved when the SLAs of all slices are fulfilled with the same level of satisfaction.

In Fig. 4, we show the average PLR of all slices as a function of cell load ρ in a two-slice network with $\kappa = 20$ and $T = 10^4$. The target PLRs are given by $r_1 = r_2 = 0.01$. As we can see, more packets are lost as the cell load ρ increases, since the queueing delay is always increased by the additional traffic. Specifically, given the target average PLR 0.01, the proposed algorithm allows the base station to serve 82% of the full load data traffic, which is 10% higher than 72% of the optimal static algorithm and 5% lower than 87% of the optimal dynamic algorithm. Therefore, the proposed algorithm can support a

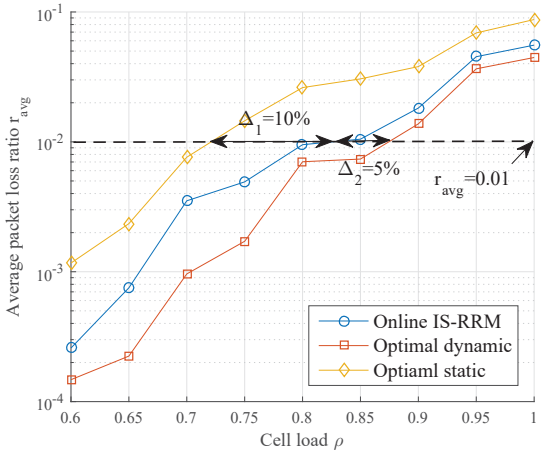

Fig. 4: Average packet loss ratio as a function of cell load with $\kappa = 20$ and $T = 10^4$.

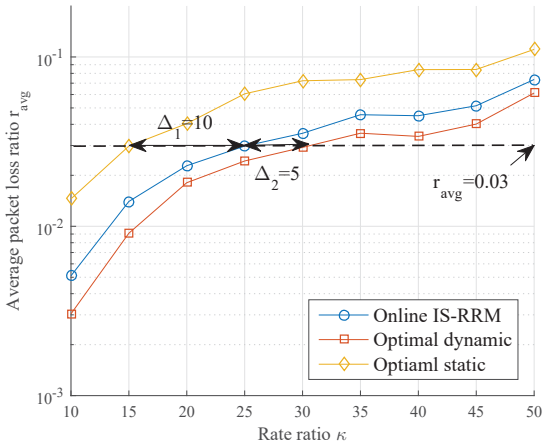

Fig. 5: Average packet loss ratio as a function of rate ratio with $\rho = 0.9$ and $T = 10^4$.

heavy loaded network with comparable performance to the optimal strategies given in hindsight.

In Fig. 5, we show the average PLR of all slices as a function of rate ratio κ in a two-slice network with $\rho = 0.9$ and $T = 10^4$. The target PLRs are given by $r_1 = r_2 = 0.03$. As we can see, more packets are lost as κ increases. The reason is that the traffic fluctuation highly increases the number of peak data rate periods, which usually lead to severe packet congestion and loss. Specifically, given the target average PLR 0.03, the proposed algorithm allows the high pattern data rate to be 25 times as large as the low pattern data rate, which is 10 larger than the ratio 15 given by the optimal static algorithm and 5 smaller than the ratio 30 given by the optimal dynamic algorithm. Therefore, the proposed algorithm can support a highly dynamic network with comparable performance to the optimal strategies given in hindsight.

V. 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 perform directional adjustment by exploiting the derivatives of well-designed loss functions. Compared with the optimal dynamic and optimal static strategies given in hindsight, the proposed online IS-RRM algorithm can provide differentiated SLA performance for each slice and achieve a comparable performance in networks with various levels of traffic loads and network dynamics.

References

[1] 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.

[2] 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.

[3] R. Kokku *et al.*, "NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.

[4] V. Sciancalepore *et al.*, "Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization," in *Proc. of IEEE INFOCOM'17*, Atlanta, GA, USA, May 2017.

[5] 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.

[6] 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, Sep. 2019.

[7] Q. Ye, W. Zhuang, S. Zhang, A. Jin, X. Shen, and X. Li, "Dynamic Radio Resource Slicing for a Two-Tier Heterogeneous Wireless Network," *IEEE Trans. Veh. Tech.*, vol. 67, no. 10, pp. 9896–9910, Oct. 2018.

[8] 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.

[9] T. Wang, S. Wang, and Z.-H. Zhou, "Machine Learning for 5G And Beyond: From Model-Based to Data-Driven Mobile Wireless Networks," *China Communications*, vol. 16, no. 1, pp. 165–175, Jan. 2019.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] J. Gondzio, "Interior Point Methods 25 Years Later," *Europ. J. Oper. Res.*, vol. 218, no. 3, pp. 587–601, May 2012.