Online Learning for Hierarchical Scheduling to Support Network Slicing in Cellular Networks

Jianhan Song, Gustavo de Veciana, Sanjay Shakkottai
ECE Department, The University of Texas at Austin

{jianhansong, deveciana, sanjay.shakkottai}@utexas.edu

ABSTRACT

We study a learning-based hierarchical scheduling framework in support of network slicing for cellular networks. This addresses settings where users and/or service classes are grouped into slices, and resources are allocated hierarchically. The hierarchy is implemented by combining a slice-level scheduler which allocates resources to slices, and flow-level schedulers within slices which opportunistically allocate resources to users/services. Optimizing the slice-level scheduler to maximize system utility is typically hard due to underlying heterogeneity and uncertainty in user channels and performance requirements. We address this by reformulating the problem as an online black-box optimization where slice-level schedulers (parameterized by a weight vector) combined with flow-level schedulers result in user/service level stochastic rewards representing performance fitness; the goal is to learn the best weight vector. We develop a bandit algorithm based on queueing cycles by building on Hierarchical Optimistic Optimization (HOO). The algorithm guides the system to improve the choice of the weight vector based on observed rewards. Theoretical analysis of our algorithm shows a sub-linear regret with respect to an omniscient genie. Finally through simulations, we show that the algorithm adaptively learns the optimal weight vectors when combined with opportunistic and/or utility-maximizing flow-level schedulers.

Keywords

Scheduling, Wireless Networks, Network Slicing, Online Learning, Bandit Algorithms

1. INTRODUCTION

The increasing complexity of cellular wireless networks has led to network slicing as a popular paradigm for resource sharing. Network slicing is a coarse resource allocation mechanism that partitions traffic flows into groups (slices), and allocates network resources (e.g. spectrum) to each of these slices. Network slicing operates alongside a finer-grain resource manager (flow-level scheduler) that allocates resources among the flows within each slice. Slicing can be used for various reasons including isolating groups from each other in the presence of traffic load fluctuations, or grouping flows with similar Quality of Service (QoS) requirements, so that flow-level schedulers can operate across groups of flows with roughly homogeneous requirements.

In this paper, we adopt a hierarchical online learning approach to network slicing that is driven by user-feedback in the form of rewards. Given a collection of slices (each slice defined through a collection of flows, and with QoS and spectrum-share requirements), we develop a slice-level scheduler (top of the hierarchy) that dynamically allocates resources to each slice based on the observed rewards from mobile users within each slice. This slice-level scheduler allocates resources by dynamically selecting weights for each slice, with these weights specifying a share of spectrum for each slice through an allocation mechanism such as a *Generalized Processor Sharing (GPS)* scheduler. Further, within each slice, a flow-level scheduler (such as the Max-Weight rule) allocates channel resources to individual flows. Thus, the reward obtained from a slicing allocation depends both on the sharing of spectrum for each slice and the individual allocations within each slice. By treating the transformation from weight selection to reward accumulation as a *blackbox function*, we build on bandit-based blackbox optimization methods to develop adaptive slicing mechanisms.

2. PROBLEM FORMULATION

We consider a queueing system with a single server (base station) serving a set of users which are further grouped into slices. Each user is associated with a stochastic packet arrival and wireless channel process. We study a hierarchical scheduling framework in which a slice-level scheduler is parameterized by a weight vector $\boldsymbol{w}$ and flow-level schedulers are pre-selected for each slice. For any choice of the weight vector $\boldsymbol{w}$, the system observes a mean reward rate associated with users' performance/utility — this mapping from weights to reward rates is represented as a function $f(\boldsymbol{w})$. See Figure 1 for an overview.

Due to complexity and dynamics of such systems, $f(\boldsymbol{w})$ is analytically hard to optimize and the problem can be better studied as a blackbox optimization. Using a multi-armed bandit framework, where the (continuous-valued) weights correspond to the arms of the bandit and the corresponding arm-rewards accrue from (noisy) user feedback, we develop an online algorithm that explores the choice of weights to adaptively optimize the blackbox function.

3. THE CHOOC ALGORITHM

We propose *Cycle-Based HOO with Clipping* (CHOOC) algorithm, a modified Hierarchical Optimistic Optimization (HOO) algorithm [1] to adaptively learn the best weight vector. CHOOC operates at the time-scale of queueing cycles

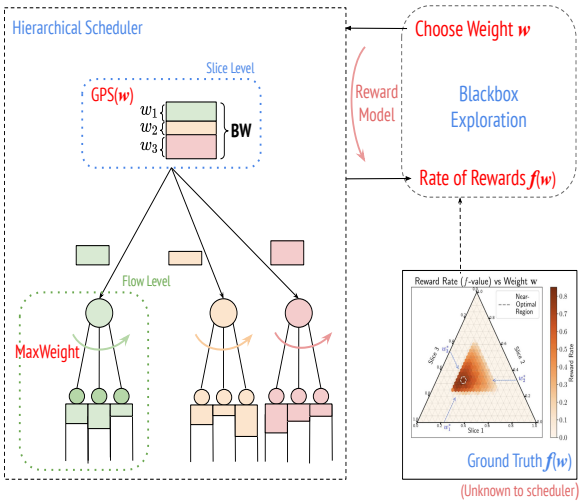

Figure 1: Illustration of a hierarchical scheduler (left) and the formulation as a blackbox optimization problem (right).

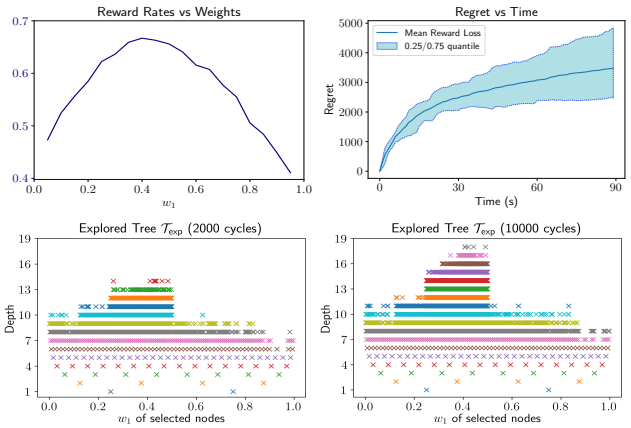

Figure 2: Simulation results on experiments in Section 4.

(idle + busy period) — the queue dynamics and rewards are conditionally independent (given $\boldsymbol{w}$) over cycles under proper assumptions, which is essential for the comparison of different arms. A single exploration sample of $f(\cdot)$ corresponds to selecting a weight vector $\boldsymbol{w}$, using these weights to allocate spectrum resources (to slices) via the associated slice-level scheduler, in turn allow predetermined flow-level schedulers to assign resources to individual users, and finally collecting the aggregate reward from active users over a queueing cycle. The weight selection mechanism underlying CHOOC involves partitioning the weight space into a binary tree, refining the estimation of $f(\boldsymbol{w})$ corresponding to each partition through collected samples, and choosing weights from partitions with the best estimated performance/rewards.

From a technical perspective, as compared to HOO we address two additional challenges: *(i) Ratio of Rewards:* Since the length of queueing cycles are random and depend on the action (the weight vector $\boldsymbol{w}$), our reward rate is described through a ratio of two random summations — reward accrued over cycles divided by the cumulative cycle lengths — thus, we need to control the associated uncertainty which does not directly fit the standard HOO model (because ratios of sums differs from the sum of ratios). *(ii) Sub-Exponential Rewards and Unstable Queues:* Unlike the sub-Gaussian reward setting of HOO, queueing cycle lengths are either sub-exponential (if $\boldsymbol{w}$ results in stable queues), or can be infinite if the queues become unstable. Thus, we need to clip cycles (i.e. truncate overly long cycles by dropping packets), but must do so with a negligible rate of clipping (to minimize drops). By properly addressing these issues, our theoretical analysis recovers a sub-linear regret, which is of the same order as HOO. See [2] for a detailed algorithm description and full theoretical results.

4. PERFORMANCE EVALUATION

We simulate our algorithm in various wireless settings, which include different slice partitions and and heterogeneous performance metrics of user packets. The experiments show our algorithm is able to locate the optimal weight after a reasonable amount of exploration.

Figure 2 exhibits a representative simulation result displaying the convergence behavior of the CHOOC algorithm. For this experiment, we simulated a simplified cellular wireless base station comprising 2 slices where each slice has 6 heterogeneous users (in terms of arrival and service rates). For the first slice, the reward of each transmitted packet is given by $(1 - \mathsf{delay} * 0.1)^+$; while for the second slice, the reward of each packet equals $\mathbb{1}_{\{\mathsf{delay}<7\}}$. The cumulative reward is defined as the sum of packet rewards over time.

The top-left panel in Figure 2 shows the (Monte Carlo-simulated) reward rate (i.e., $f(\boldsymbol{w})$). The optimum is roughly $w_1 = 0.42$. We then run CHOOC for 10k cycles. In the bottom panels of Figure 2, we show how the "explored tree" of CHOOC evolves from cycle index $n = 2$k to 10k, where each dot in the scatter plots represents a weight selection at the corresponding depth. As expected, the tree grows deeper around the optimum, implying that CHOOC is focusing on exploring near-optimal weights. Convergence is further verified by the time-vs-regret plot, which shows a sub-linear growth.

See [2] for complete experiment results and additional insights/conclusions.

Acknowledgements

This work was supported by NSF grants CNS-1731658, CNS-1718089 and CNS-1910112, Army Futures Command Grant W911NF1920333, and the Wireless Networking and Communications Group Industrial Affiliates Program.

5. REFERENCES

[1] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12(5), 2011.

[2] Jianhan Song, Gustavo de Veciana, and Sanjay Shakkottai. Online learning for hierarchical scheduling to support network slicing in cellular networks. *Performance Evaluation*, page 102237, 2021.