Robust Wireless Scheduling under Arbitrary Channel Dynamics and Feedback Delay (Invited Paper)

Jiatai Huang
IIIS, Tsinghua University
hjt18@mails.tsinghua.edu.cn

Longbo Huang
IIIS, Tsinghua University
longbohuang@tsinghua.edu.cn

***Abstract*—Designing efficient scheduling algorithms is crucial to the development of modern wireless networks. In this paper, we study a wireless network model consisting of one central base-station and K mobile users. Each time the base-station can simultaneously transmit to $1 \leq M \leq K$ users. The channel states change over time adversarially, and the feedback of transmission outcome can experience arbitrary delays. The objective of the base-station is to search for a policy to maximize the overall transmission success rate. We propose a scheduling algorithm named `Banker-OMD-Scheduling` for this setting, based on a recent banker online mirror descent technique [1]. We show that `Banker-OMD-Scheduling` guarantees that the total regret over a finite time horizon T is $O(\sqrt{MK}(\sqrt{T}+\sqrt{D\log D}))$ where D is the total feedback delay.**

***Index Terms*—wireless, scheduling, online bandit optimization, delayed feedback**

I. Introduction

In the past decade, due to the rapid development of mobile technology, wireless networks have become an irreplaceable part of the world-wide communication infrastructure. Among the many technical challenges, scheduling has been one of the most fundamental problems in wireless network control and has been extensively studied, e.g., [2], [3], [4], [5] and [6]. However, existing results often focus on stochastic settings and assume instantaneous feedback for the transmissions. Thus, they do not directly apply to scenarios where the system dynamics are non-stationary.

In this paper, we study a model for wireless network under adversarial channel dynamics with delayed feedback. Specifically, we consider the following discrete-time model. There is a base-station and K mobile users. In each time step, the base-station can choose $1 \leq M \leq K$ users and transmit to them. The channel states can change over time *arbitrarily*. The base-station has very limited knowledge on the current channel states, and needs to schedule only based on *bandit* feedback about the amount of data transmitted. That is, the base-station can only know the number of bits transmitted to the M selected users after the decision has been made, and it gets no information about the links of the remaining $K - M$ users. Moreover, due to dynamics, the transmission feedback may arrive with delays. Furthermore, the base-station is given no knowledge about the delay length of each time step until this feedback arrives. The objective of the base-station is to find a scheduling policy to decide which users to transmit and optimize the total throughput.

This setting is very general and can be used to model many different communication processes built on different physical media. However, it is also challenging due to the arbitrary channel dynamics and feedback delay. To address the difficulties, we propose a scheduling algorithm based on the `Banker-OMD` framework recently developed in [1], and design `Banker-OMD-Scheduling`. We prove that `Banker-OMD-Scheduling` achieves an $\tilde{O}(\sqrt{T}+\sqrt{D})$ total regret under any finite time horizon length T.[1] Here D denotes the total feedback delay of all transmissions over the T time steps, and the regret is defined as the difference between total amount of data transmitted under our algorithm and an optimal static policy serving a fixed subset of M users. Note that under moderate delays, for example, $D = O(T)$, our algorithm achieves $\tilde{O}(\sqrt{T})$ total regret, which means that for a sufficiently large time horizon length, the relative overhead of our scheduling is vanishing.

Online learning based scheduling algorithms have been proposed and studied in the scheduling literature, e.g., [7], [8] and [9], where regret is also adopted as the primary performance metric. Compared to these prior results, our work differs in that we allow serving multiple users simultaneously under arbitrary channel conditions, and our algorithm is robust to feedback delay. Our formulation is similar to the M-set adversarial semi-bandit problem [10], where there is no feedback delay and there have been algorithms achieving $O(\sqrt{MKT})$ total regret, e.g., [10]. In the terms of bandit optimization problems, each user in our setting is a bandit *arm*, the base-station is an *agent* that needs to select a subset of arms of size M at each time step. Another previous online optimization problem similar to our setting is the adversarial MAB problem with feedback delays, which is a special case of our setting where $M = 1$, and there have been algorithms achieving $O(\sqrt{KT}+\sqrt{D\log K})$ total regret [11]. Hence, our setting can be viewed as a hybrid generalization of these two aforementioned online optimization problems.

[1]Throughout this paper, we use the $\tilde{O}$ notation to suppress poly-logarithmic factors in T.

ISBN 978-3-903176-43-0

II. Notation

We use $[K]$ to denote the set $\{1, 2, \ldots, K\}$. $\mathbf{0}$ denotes the zero vector. Let f be any strictly convex function defined on some convex $A \subseteq \mathbb{R}^K$. For any $x, y \in A$, if $\nabla f(x)$ exists, we use $D_f(y, x) := f(y) - f(x) - \langle \nabla f(x), y - x \rangle$ to denote the Bregman divergence between y and x induced by f. We use

$$f^*(y) := \sup_{x \in \mathbb{R}^K} \{\langle y, x \rangle - f(x)\}$$

to denote the convex conjugate of f. Define $\Delta^{[M,K-1]}$ to be the following set

$$\Delta^{[M,K-1]} := \left\{ x \in \mathbb{R}_+^K : \sum_{i=1}^K x_i = M \right\}.$$

Unless stated otherwise, for any strictly convex function f, we define $\bar{f}$ as follows: for all $x \in \mathbb{R}^K$,

$$\bar{f}(x) := \begin{cases} f(x) & x \in \Delta^{[M,K-1]} \\ \infty & x \notin \Delta^{[M,K-1]}. \end{cases}$$

We say a convex function f is *Legendre* if

1) $int(A)$ is non-empty;
2) f is differentiable on $int(A)$;
3) $\lim_{n \to \infty} \|\nabla f(x_n)\|_2 \to \infty$ for any sequence $(x_n)_{n=1}^\infty$ in $int(A)$ converging to some $x \in \partial\, int(A)$.

III. Problem Setting

We consider a slotted-time system containing a central base-station and K users. We assume the base-station can serve $1 \leq M \leq K$ users at a time, by performing data transmission over the M channels to the users. Denote by $\mathcal{A}$ the set of all feasible choices of subset of users to serve, i.e., $\mathcal{A} := \{S \subseteq [K] : |S| = M\}$.

The state of each channel at each time slot is assumed to vary *arbitrarily*. Specifically, denote by $r_{t,i}$ the amount of data that will be transmitted to user i if the base-station decides to serve user i at time t. $r_{t,i}$ is determined by the environment at the beginning of time slot t, simultaneously when the base-station makes its decision (that is, to choose a subset of users of size M to serve). We assume all $r_{t,i}$ values are in $[0, 1]$ and make no further assumptions on $r_{t,i}$'s. i.e., the environment can be arbitrary.

Denote by S_t the set of users the base-station chooses to serve at time t, and let $A_{t,i}$ be the indicator of whether user i is served at time slot t, i.e., $A_{t,i} := \mathbb{1}[i \in S_t]$. For any time horizon of finite length T, the total amount of data successfully transmitted is

$$U_T := \sum_{t=1}^T \sum_{i \in S_t} r_{t,i} = \sum_{t=1}^T \sum_{i=1}^K A_{t,i} r_{t,i}.$$

For each time index t, the values of $r_{t,\cdot}$ are not known to the base-station. Only values of the served users at time t will be revealed to the base-station eventually. To be specific, there is a delay length sequence $(d_t)_{t=1}^\infty$, $d_t \in \mathbb{N}$, chosen by the environment beforehand (unknown to the base-station a-priori). Suppose at time t the base station serves users in S_i. Then, at the end of time slot $t + d_t$, the base-station will receive a vector

$$\tilde{r}_t = (A_{t,1} r_{t,1}, A_{t,2} r_{t,2}, \ldots, A_{t,K} r_{t,K})$$

from the environment.

The design objective of a scheduling policy is to maximize U_T, the total data transmitted. Since in many cases we are dealing with an indefinite time horizon rather than optimize over a fixed finite T, it is more convenient to introduce the classical performance metric called the **pseudo-regret** (referred to as regret below for convenience):

Definition 1 (Pseudo-regret of a Scheduling Policy)**.** *We define*

$$\mathfrak{R}_T := \max_{S \in \mathcal{A}} \mathbb{E}[\sum_{t=1}^T \sum_{i \in S} r_{t,i} - \sum_{t=1}^T \sum_{i \in S_t} r_{t,i}] \tag{1}$$

to be the pseudo-regret of an MAB algorithm, where the expectation is taken with respect to both the scheduling policy's internal randomness and randomness from the environment.

It is easy to see that maximizing U_T is equivalent to minimizing $\mathcal{R}_T$. Following the convention in bandit optimization problems, we are asking for a scheduling policy achieving $o(T)$ total regret.

Our formulation can be viewed as a bandit optimization problem, where each user in our setting is a bandit *arm*, the base-station is an *agent* that needs to select a subset of arms of size M at each time step, and the *reward* received is total amount of data transmitted. In this regard, our model generalizes the M-set adversarial semi-bandit problem [10], where there is no feedback delay, i.e., all d_t's are equal to 0, and there have been algorithms achieving $O(\sqrt{MKT})$ total regret, e.g., [10], and the adversarial MAB problem with feedback delays, which is a special case of our setting where $M = 1$, and there have been algorithms achieving $O(\sqrt{KT} + \sqrt{D \log K})$ total regret, e.g., [11].

Follow the convention of online optimization problems with delayed feedback, we will heavily use the following two quantities in derived regret bounds, the individual delay upper bound $d := \max_{t \in [T]} d_t$ and total delay $D := \sum_{t=1}^T d_t$.

Unless stated otherwise, we will use

$$\mathcal{F}_t = \sigma\left(S_1, \ldots, S_t, \tilde{r}_1 \mathbb{1}[1 + d_1 \leq t], \ldots, \tilde{r}_t \mathbb{1}[t + d_t \leq t]\right)$$

to denote the filtration of σ-algebra when studying random quantities indexed by time.

Below, we will present our algorithms and results. For ease of presentation, all proofs are presented in the appendix.

IV. Preliminaries

Our proposed algorithm for this wireless scheduling problem is based upon a recent novel Online Mirror Descent (OMD) framework called `Banker-OMD` [1], which is a natural generalization of classical OMD method to robustly handle feedback delays in online optimization problems. In this section we present a brief overview of the classical OMD method and `Banker-OMD`, and apply them to solving the

M-set Semi-Bandit problem. The algorithms here will serve as the building block for designing the scheduling algorithm.

A. Online Mirror Descent for Non-delayed M-set Semi-Bandit

Mirror descent is a concept originated in classical optimization [12]. To apply mirror descent in online bandit optimization problems, the typical approach is to design an unbiased estimator for the true reward (or loss) vector, which is r_t in our setting. Then, regard this loss estimator as a "gradient" and use it to take a gradient descent step, where the update is performed in the dual space induced by some Legendre function Ψ rather than in the primal space. Algorithm 1 below presents our algorithm for solving the M-set Semi-Bandit problem based on OMD.

Algorithm 1: Online Mirror Descent for Non-delayed M-set Semi-Bandit

Input: Number of arms K, Number of arms to pull at each time step M, time horizon length T, Legendre function $\Psi : \mathbb{R}_+^K \to \mathbb{R}$, initial mixed action $x_1 \in \Delta^{[M,K-1]}$, action scales $\sigma_1, \ldots, \sigma_T$

Output: A sequence of actions $S_1, S_2, \ldots, S_T \in \mathcal{A}$

1 **for** $t = 1, 2, \ldots, T$ **do**
2 Let p_t be a distribution on $\mathcal{A}$ such that $x_{t,i} = \mathbb{P}_{S\sim p_t}[i \in S]$ for all $i \in [K]$;
3 Sample $S_t \in \mathcal{A}$ according to p_t, serve the M users in S_t, receive the feedback vector $\tilde{r}_t$;
4 Compute the importance sampling loss estimate vector $\tilde{l}_t$ by $\tilde{l}_{t,i} \leftarrow \frac{(1-\tilde{r}_{t,i})A_{t,i}}{x_{t,i}}$;
5 $x_{t+1} \leftarrow \nabla\bar{\Psi}^*(\nabla\Psi(x_t) - \frac{1}{\sigma_t}\tilde{l}_t)$;
6 **end**

The choice of $\tilde{l}_t$ in Algorithm 1 (Line 4) guarantees that $\mathbb{E}[\tilde{l}_t \mid \mathcal{F}_{t-1}] = 1 - r_t$. Recall that in our problem setting, all reward values picked by the environment are $[0, 1]$-bounded, equivalently we can regard serving user i at time t as an action that incurs a loss of $1 - r_{t,i}$, and in this sense the $\tilde{l}_t$ in Line 4 is an unbiased estimate for the true loss vector $l_t := 1 - r_t$.

To establish the performance of Algorithm 1, we make use of the following lemma, which concerns the regret incurred in a single time-step to the OMD update happened in that time-step. This lemma is a standard result and can be found in many OMD textbooks and literatures (e.g., [13] and Chapter 28 of [14]).

Lemma 1. *For any $\sigma_t > 0$, $x_t, y \in \Delta^{[M,K-1]}$, $l_t \in \mathbb{R}_+^K$ and Legendre function $\Psi : \mathbb{R}_+^K \to \mathbb{R}$, we have*

$$\langle l_t, x_t - y\rangle \le \sigma_t D_\Psi(y, x_t) - \sigma_t D_\Psi(y, z_t) + \sigma_t D_\Psi(x, \tilde{z}_t) \quad (2)$$

where

$$z_t = \operatorname*{arg\,min}_{x' \in \Delta^{[M,K-1]}} \langle l_t, x'\rangle + \sigma_t D_\Psi(x', x_t), \quad (3)$$

$$\tilde{z}_t = \operatorname*{arg\,min}_{x' \in \mathbb{R}_+^K} \langle l_t, x'\rangle + \sigma_t D_\Psi(x', x_t), \quad (4)$$

or equivalently,

$$z_t = \nabla\bar{\Psi}^*(\nabla\Psi(x_t) - \frac{1}{\sigma_t} l_t), \quad \tilde{z}_t = \nabla\Psi^*(\nabla\Psi(x_t) - \frac{1}{\sigma_t} l_t). \quad (5)$$

We see that Algorithm 1 is effectively obtained by taking an OMD step on the last mixed action x_t using $\tilde{l}_t$ as the gradient and $1/\sigma_t$ as the step size, and then taking the computed z_t in Lemma 1 as the next mixed action. This procedure naturally leads to a telescoping sum regret upper-bound when all σ_t's are set to the same constant. Then, by properly choosing the constant we can achieve $O(\sqrt{T})$ (factors in K and M omitted) total regret.

By summing up the inequality in Lemma 1, we get the following regret bound for Algorithm 1:

Theorem 2. *The total regret of Algorithm 1 satisfies*

$$\mathfrak{R}_T \le \sup_{y\in\mathcal{A}} \mathbb{E}\Big[\sum_{t=1}^{T} \{\sigma_t D_\Psi(y, x_t) - \sigma_t D_\Psi(y, x_{t+1}) + \sigma_t D_\Psi(x_t, \tilde{z}_t)\}\Big]$$

where $\tilde{z}_t = \nabla\Psi^*(\nabla\Psi(x_t) - \frac{1}{\sigma_t}\tilde{l}_t)$.

In particular, when $\Psi(x) = -\sum_{i=1}^K 2\sqrt{x_i}$, $x_1 = (M/K, \ldots, M/K)$ and $\sigma_1 = \cdots \sigma_T = \sqrt{T}$, we have the following regret bound for Algorithm 1:

Theorem 3. *If Algorithm 1 is running with $\Psi(x) = -\sum_{i=1}^K 2\sqrt{x_i}$, $x_1 = (M/K, \ldots, M/K)$ and $\sigma_1 = \cdots \sigma_T = \sqrt{T}$, then we have*

$$\mathfrak{R}_T \le O(\sqrt{MKT}).$$

Remark. Under this particular choice of Ψ and x_1, Algorithm 1 and our novel Algorithm 3 presented in Section V become identical when running on problem instances without delay. Therefore Theorem 3 is a direct corollary of Theorem 6 in Section V, our main regret bound for Algorithm 3.

B. The `Banker-OMD` Framework

A key observation in Theorem 2 is that when $\sigma_1, \ldots, \sigma_T$ are all chosen to be a constant, the sum $\sum_{t=1}^T \{\sigma_t D_\Psi(y, x_t) - \sigma_t D_\Psi(y, x_{t+1})\}$ in Theorem 2 will become a telescoping sum and equal to $\sigma_1 D_\Psi(y, x_1) - \sigma_1 D_\Psi(y, x_{T+1})$. However, in order to compute z_t, we need to compute the loss estimator $\tilde{l}_t$ first, which relies on the observed feedback vector $\tilde{r}_t$. In the general case where there are feedback delays, i.e., d_t can take a positive value, we will not be able to compute z_t in time at the end of t-th time slot, and therefore unable to directly set x_{t+1} to z_t.

To tackle this drawback of classical OMD method, [1] proposed the `Banker-OMD` framework, which is capable of naturally proceeding even when there are absent feedback due to delay. `Banker-OMD` builds upon the core observation that Lemma 1 applies to *any* mixed action sequence $x_1, \ldots, x_T$ and action scale sequence $\sigma_1, \ldots, \sigma_T$ and gives a valid upper-bound for the pre-expectation total regret $\sum_{t=1}^T \langle \tilde{l}_t, x_t - y\rangle$.

Under Banker-OMD, the task of getting a tight regret bound reduces to designing a policy to choose x_t's and σ_t's to make $\sum_{t=1}^{T}\{\sigma_t D_\Psi(y,x_t)-\sigma_t D_\Psi(y,z_t)\}$ small. In Banker-OMD the $\sigma_t D_\Psi(y,x_t)$ term is called a "withdrawal cost," for it is a valid upper-bound for (besides the immediate cost term $\sigma D_\Psi(x_t,\tilde{z}_t)$) the loss incurred due to taking the action x_t, without the need of knowing the action feedback $\tilde{r}_t$. After receiving $\tilde{r}_t$, we can refine this upper-bound to $\sigma_t D_\Psi(y,x_t)-\sigma_t D_\Psi(y,z_t)$, where the term $-\sigma_t D_\Psi(y,z_t)$ is called the "saving term" and is some resource we can leverage only after receiving the action feedback r_t. From this reasoning, the task of getting a tight regret bound can be further reduced to *properly covering the withdrawal cost of a new action with savings at hand.*

The next key observation of Banker-OMD is that, suppose at some moment we have available saving terms $\sigma_1 D_\Psi(y,z_1),\ldots,\sigma_h D_\Psi(y,z_h)$ at hand, then a new action obtained from "convex combination in the dual space" over available z_t's always has its withdrawal cost properly covered. This fact is formally stated in the following Lemma 4, which is an adapted version of Lemma 2 of [1], extending the feasible action set from $\Delta^{[1,K-1]}$ in Lemma 2 of [1] to $\Delta^{[M,K-1]}$.

Lemma 4. *For any $h\geq 1$, $z_1,\ldots,z_h\in\Delta^{[M,K-1]}$, $\sigma_1,\ldots,\sigma_h>0$ and Legendre convex function $\Psi:\mathbb{R}_+^K\to\mathbb{R}$, let $\sigma=\sum_{i=1}^{h}\sigma_i$ and*

$$x=\nabla\bar{\Psi}^*\Big(\sum_{i=1}^{h}\frac{\sigma_i}{\sigma}\nabla\Psi(z_i)\Big),\tag{6}$$

then we have $\sigma D_\Psi(y,x)\leq\sum_{i=1}^{h}\sigma_i D_\Psi(y,z_i)$ for any $y\in\Delta^{[M,K-1]}$.

The Banker-OMD framework in [1] works as follows. Lemma 4 states that if at the beginning of time t we have available saving terms $\sigma_{t_1}D_\Psi(y,z_{t_1}),\ldots,\sigma_{t_h}D_\Psi(y,z_{t_h})$, i.e., feedback of actions at time $t_1,\ldots,t_h$ have all arrived and $z_{t_1},\ldots,z_{t_h}$ have not been utilized in any form. Let $\sigma=\sum_{i=1}^{h}\sigma_{t_i}$, then we are allowed to pick a new action x_t with scale $\sigma_t=\sigma$ specified by $x_t=\nabla\bar{\Psi}^*(\sum_{i=1}^{h}\frac{\sigma_{t_i}}{\sigma_t}\nabla\Psi(z_{t_i}))$. The withdrawal term introduced by playing this x_t at scale σ_t is then guaranteed to be covered by those h saving terms.

It can happen that in a time slot, savings can fall short, e.g., no feedback has arrived, but we still need to make a decision in run-time. In this case, we can use a scale $\sigma_t>\sigma$. To do so, let $b_t=\sigma_t-\sigma$. Then, consider an $x_0\in\Delta^{[M,K-1]}$. We can add two new terms $+b_tD_\Psi(y,x_0)-b_tD_\Psi(y,x_0)$ (sum to zero) to the current upper-bound for the total regret. Then, we can regard the minus-signed term $-b_tD_\Psi(y,x_0)$ as an additional saving term, and use it together with the h available previous saving terms to form the new x_t by $x_t=\nabla\bar{\Psi}^*(\sum_{i=1}^{h}\frac{\sigma_{t_i}}{\sigma_t}\nabla\Psi(z_{t_i})+\frac{b_t}{\sigma_t}\nabla\Psi(x_0))$. Lemma 4 then asserts the net effect of playing x_t of scale σ_t is to use up the h saving terms and add a $+b_tD_\Psi(y,x_0)$ term into the upper-bound. This operation can be interpreted as: when the total savings are not enough for the new action we want to form, we can *invest* in some x_0 to make up the difference and proceed.

Based on the above banker idea and the Banker-OMD algorithm from [1], we design Algorithm 2 Banker-OMD-M for our delayed M-set semi-bandit problem. We emphasize that Banker-OMD-M is a general algorithm *framework*, it offers great flexibility in algorithm design, i.e., the regularizer Ψ and the scales σ_t,b_t,m_t. In Banker-OMD-M, $\mathfrak{S}$ is a policy to choose action scales σ_t, in practice σ_t is chosen to a function of the current time index t and statistics of experienced delays. $\mathfrak{P}$ is the policy for deciding the desired σ_t based on available savings and possible new investment. At the beginning of each time step t, $\mathfrak{P}$ needs to output t non-negative numbers $b_t,m_{t,1},\ldots,m_{t,t-1}$ such that $b_t+\sum_{i=1}^{t-1}m_{t,i}=\sigma_t$ and $m_{t,i}\leq v_i$ for all $1\leq i\leq t-1$.

The difference between Banker-OMD-M and Banker-OMD is that the feasible action set is $\Delta^{[M,K-1]}$ rather than the unit simplex, and we need to appropriately sample actions (Line 13) and compute loss estimates (Line 17). Similar to the original Banker-OMD framework, the key feature of Banker-OMD-M is that it explicitly maintains an OMD-style upper-bound for cumulative pseudo-regret. At any time during the execution, its internal state variables directly give a regret upper-bound stated in the following Theorem 5, which adapts Theorem 3 of [1] and extends the feasible action set to $\Delta^{[M,K-1]}$.

Theorem 5 (Banker-OMD-M Regret Bound)**.** *For Algorithm 2, at the end of any time T, for any $y\in\Delta^{[M,K-1]}$, we have*

$$\begin{aligned}&\sum_{t=1}^{T}\langle\tilde{l}_t,x_t-y\rangle\\&\leq B_T\cdot D_\Psi(y,x_0)+\sum_{t=1}^{T}\sigma_t D_\Psi(x_t,\tilde{z}_t)-\sum_{t=1}^{T}v_t D_\Psi(y,z_t)\end{aligned}\tag{7}$$

where $B_T,\tilde{l}_1,\ldots,\tilde{l}_t,x_1,\ldots,x_t,z_1,\ldots,z_t$ are variable values in Algorithm 2 at the end of time T, $\tilde{z}_t=\nabla\Psi^*(\nabla\Psi(x_t)-\frac{1}{\sigma_t}\tilde{l}_t)$ for any $1\leq t\leq T$.

V. The Banker-OMD-Scheduling Algorithm

We are now ready to present our algorithm for the wireless scheduling problem with channel variation and feedback delay. The proposed algorithm Banker-OMD-Scheduling (Algorithm 3) is an application of the Banker-OMD-M framework introduced in Section IV-B using the regularizer capable to achieve $O(\sqrt{MKT})$ total regret in the non-delayed setting, described in Section IV-A.

Comparing to the general framework Algorithm 2, Banker-OMD-Scheduling chooses a specific regularizer $\Psi(x)=-2\sum_{i=1}^{K}\sqrt{x_i}$, which is known as the 1/2-Tsallis entropy function [13], and default investment option $x_0=(M/K,\ldots,M/K)$. In Banker-OMD-Scheduling, the subroutine used to make plan of integrating available savings and possible new investment to the desired new action

Algorithm 2: `Banker-OMD-M`

Input: Number of arms K, number of arms to pull at each time step M, regularizer Ψ, default investment option $x_0 \in \Delta^{[M,K-1]}$, subroutine to pick action scales $\mathfrak{S}$, subroutine to pick portfolio $\mathfrak{P}$

Output: A sequence of actions $S_1, S_2, \ldots \in \mathcal{A}$

1 $B_0 \leftarrow 0$; // $B_\cdot$ will be maintained as the total investment to x_0
2 **for** $t = 1, 2, \ldots$ **do**
3 $a_t \leftarrow 0$; // a_t indicates whether time t's feedback has arrived
4 $h_t \leftarrow (S_1, \ldots, S_{t-1}, a_1, \ldots, a_{t-1}, r_1, \ldots, r_{t-1}, \sigma_1, \ldots, \sigma_{t-1}, v_1, \ldots, v_{t-1})$; // history to up time $t-1$
5 $\sigma_t \leftarrow \mathfrak{S}(t, h_t)$; // decide the scale for the new action
6 $v_t \leftarrow \sigma_t$; // v_t is the current coefficient of the minus-signed $D_\Psi(y, z_t)$ term
7 $b_t, m_{t,1}, \ldots, m_{t,t-1} \leftarrow \mathfrak{P}(t, h_t, \sigma_t)$; // determine to use how much volume of available saving terms to form x_t
8 $x_t \leftarrow \nabla\bar{\Psi}^*(\frac{1}{\sigma_t}\sum_{i=1}^{t-1} m_{t,i}\nabla\Psi(z_i) + \frac{b_t}{\sigma_t}\nabla\Psi(x_0))$;
9 $B_t \leftarrow B_{t-1} + b_t$;
10 **for** $i = 1$ **to** $t-1$ **do**
11 $v_i \leftarrow v_i - m_{t,i}$; // spend $m_{t,i}$ units of the saving term $D_\Psi(y, z_i)$
12 **end**
13 Let p_t be a distribution on $\mathcal{A}$ such that $x_{t,i} = \mathbb{P}_{S\sim p_t}[i \in S]$ for all $i \in [K]$;
14 Sample $S_t \in \mathcal{A}$ according to p_t, serve the M users in S_t;
15 **for** *upon receiving each new feedback* $(s, \tilde{r}_s)$ **do**
16 Compute the importance sampling loss estimate vector $\tilde{l}_s$ by $\tilde{l}_{s,i} \leftarrow \frac{(1-\tilde{r}_{s,i})A_{s,i}}{x_{s,i}}$;
17 $z_s \leftarrow \nabla\bar{\Psi}^*(\nabla\Psi(x_s) - \frac{1}{\sigma_s}\tilde{l}_s)$;
18 $a_s \leftarrow 1$; // the saving term $-\sigma_s D_\Psi(y, x_s)$ becomes available
19 **end**
20 **end**

scale σ_t is the following procedure `GreedyPick` (simply ported from [1]), which chooses $m_{t,i}$'s greedily as large as possible and minimizes the new b_t:

Algorithm 3 maintains $\mathfrak{d}_t$, the number of feedback that have not arrived at the beginning of time t, and computes $\mathfrak{D}_t = \sum_{s=1}^{t} \mathfrak{d}_s$ to keep track of the cumulative experienced delay up to time t at run-time. Algorithm 3 then uses the action scale $\sigma_t = (\sqrt{\frac{1}{t}} + \mathfrak{d}_t\sqrt{\frac{\ln(\mathfrak{D}_t+1)}{\mathfrak{D}_t}})^{-1}$.

Remark. The last degree of freedom in Algorithm 3 is how to construct of the distribution p_t in Line 13 and how to implement a corresponding sampling scheme in Line 14. [15], [16] showed that sampling according to mean vector x_t can be

Algorithm 3: `Banker-OMD-Scheduling`

Input: Total number of users K, number of users to serve at each time step M

Output: A sequence of actions $S_1, S_2, \ldots \in \mathcal{A}$

1 Let Ψ be the function $\Psi(x) = -2\sum_{i=1}^{K}\sqrt{x_i}$;
2 $\mathfrak{D}_0 \leftarrow 0$;
3 **for** $t = 1, 2, \ldots$ **do**
4 $\mathfrak{d}_t \leftarrow$ the number of previous actions whose feedback has not arrived;
5 $\mathfrak{D}_t \leftarrow \mathfrak{D}_{t-1} + \mathfrak{d}_t$;
6 $\sigma_t \leftarrow (\sqrt{\frac{1}{t}} + \mathfrak{d}_t\sqrt{\frac{\ln(\mathfrak{D}_t+1)}{\mathfrak{D}_t}})^{-1}$;
7 $v_t \leftarrow \sigma_t$; $a_t \leftarrow 0$;
8 $b_t, m_{t,1}, \ldots, m_{t,t-1} \leftarrow$ GreedyPick$(\sigma_t, v_1, \ldots, v_{t-1}, a_1, \ldots, a_{t-1})$;
9 $x_t \leftarrow \nabla\bar{\Psi}^*(\frac{1}{\sigma_t}\sum_{i=1}^{t-1} m_{t,i}\nabla\Psi(z_i))$; // equivalent to choosing $x_0 = \nabla\bar{\Psi}^*(\mathbf{0}) = (M/K, \ldots, M/K)$
10 **for** $i = 1$ **to** $t-1$ **do**
11 $v_i \leftarrow v_i - m_{t,i}$;
12 **end**
13 Let p_t be a distribution on $\mathcal{A}$ such that $x_{t,i} = \mathbb{P}_{S\sim p_t}[i \in S]$ for all $i \in [K]$;
14 Sample $S_t \in \mathcal{A}$ according to p_t, serve the M users in S_t;
15 **for** *upon receiving each new feedback* $(s, \tilde{r}_s)$ **do**
16 Compute the importance sampling loss estimate vector $\tilde{l}_s$ by $\tilde{l}_{s,i} \leftarrow \frac{(1-\tilde{r}_{s,i})A_{s,i}}{x_{s,i}}$;
17 $z_s \leftarrow \nabla\bar{\Psi}^*(\nabla\Psi(x_s) - \frac{1}{\sigma_s}\tilde{l}_s)$;
18 $a_s \leftarrow 1$;
19 **end**
20 **end**

Procedure GreedyPick (from [1])

Params: Time index t; Current available saving coefficients $v_1, \ldots, v_{t-1}$; Availability flags $a_1, \ldots, a_{t-1}$; Target scale σ_t.

Output: $b_t, m_{t,1}, \ldots, m_{t,t-1}$.

1 $\forall 1 \le i \le t-1 : m_{t,i} \leftarrow 0$;
2 $b_t \leftarrow \sigma_t$;
3 **for** $i = 1$ **to** $t-1$ **do**
4 **if** $a_i = 1$ **then**
5 **if** $v_i \le b_t$ **then**
6 $m_{t,i} \leftarrow v_i$; $b_t \leftarrow b_t - v_i$;
7 **else**
8 $m_{t,i} \leftarrow b_t$; $b_t \leftarrow 0$;
9 **break**;
10 **end**
11 **end**
12 **end**

achieved in $O(K^2)$ time, [17] further proposed an $O(K \log K)$ sampling algorithm.

The regret bound for Algorithm 3 is summarized in the following theorem.

Theorem 6. *For the wireless scheduling problem under arbitrary channel dynamics and total feedback delay D, over any time horizen length T, `Banker-OMD-Scheduling` guarantees that*

$$\mathfrak{R}_T \leq O(\sqrt{MK}(\sqrt{T} + \sqrt{D \log D})).$$

To our knowledge, this is the first work achieving $\tilde{O}(\sqrt{T} + \sqrt{D})$ regret in this delayed M-set semi-bandit optimization problem and our scheduling problem. In most real-world scenarios, the individual delay d_t is $o(t^\alpha)$ for any $\alpha > 0$, under which Algorithm 3 guarantees that the relative overhead is vanishing as $T \to \infty$.

VI. Conclusion

In this paper, we present the `Banker-OMD-Scheduling` algorithm for solving the scheduling problem under arbitrary channel state dynamics and delayed feedback in downlink transmission. `Banker-OMD-Scheduling` is designed based on a recent banker online mirror descent framework in [1]. We prove that `Banker-OMD-Scheduling` achieves a sublinear regret and allows efficient implementation.

Acknowledgement

This work is supported in part by the Technology and Innovation Major Project of the Ministry of Science and Technology of China under Grant 2020AAA0108400 and 2020AAA0108403.

References

[1] J. Huang and L. Huang, "Banker online mirror descent," *arXiv preprint arXiv:2106.08943*, 2021.

[2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conference on Decision and Control*. IEEE, 1990, pp. 2130–2132.

[3] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1489–1501, 2006.

[4] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 841–854, 2010.

[5] R. Singh and P. Kumar, "Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 127–142, 2018.

[6] K. Chen and L. Huang, "Timely-throughput optimal scheduling with prediction," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2457–2470, 2018.

[7] Q. Liang and E. Modiano, "Minimizing queue length regret under adversarial network models," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 1, pp. 1–32, 2018.

[8] T. Choudhury, G. Joshi, W. Wang, and S. Shakkottai, "Job dispatching policies for queueing systems with unknown service rates," *arXiv preprint arXiv:2106.04707*, 2021.

[9] W.-K. Hsu, J. Xu, X. Lin, and M. R. Bell, "Integrate learning and control in queueing systems with uncertain payoffs," *Purdue University, available at https://engineering. purdue. edu/% 7elinx/papers. html, Tech. Rep*, 2018.

[10] J.-Y. Audibert, S. Bubeck, and G. Lugosi, "Regret in online combinatorial optimization," *Mathematics of Operations Research*, vol. 39, no. 1, pp. 31–45, 2014.

[11] J. Zimmert and Y. Seldin, "An optimal algorithm for adversarial bandits with arbitrary delays," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 3285–3294.

[12] A. S. Nemirovskij and D. B. Yudin, "Problem complexity and method efficiency in optimization," 1983.

[13] J. D. Abernethy, C. Lee, and A. Tewari, "Fighting bandits with a new kind of smoothness," *Advances in Neural Information Processing Systems*, vol. 28, pp. 2197–2205, 2015.

[14] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[15] M. K. Warmuth and D. Kuzmin, "Randomized online pca algorithms with regret bounds that are logarithmic in the dimension," *Journal of Machine Learning Research*, vol. 9, no. Oct, pp. 2287–2320, 2008.

[16] D. Suehiro, K. Hatano, S. Kijima, E. Takimoto, and K. Nagano, "Online prediction under submodular constraints," in *International Conference on Algorithmic Learning Theory*. Springer, 2012, pp. 260–274.

[17] J. Zimmert, H. Luo, and C.-Y. Wei, "Beating stochastic and adversarial semi-bandits optimally and simultaneously," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7683–7692.

Appendix

Technical Proofs

We present the detailed proofs in the appendix. To begin with, we review some useful properties of Legendre functions.

Lemma 7. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set, $f : \mathcal{C} \to \mathbb{R}$ be a Legendre function. Then,*

1) *∇f is a bijection between $int(\mathcal{C})$ and $int(dom(f^*))$ with the inverse $(\nabla f)^{-1} = \nabla f^*$;*
2) *$D_f(y, x) = D_{f^*}(\nabla f(x), \nabla f(y))$ for all $x, y \in int(\mathcal{C})$;*
3) *the convex conjugate f^* is Legendre.*

The proof for Lemma 7 can be found in many convex analysis textbook, e.g., Chapter 26 of [14]. We are now ready to give proofs for lemmas and theorems in the main text.

A. Proof of Lemma 1

Lemma 1 is a basic property of mirror descent update steps, the proof is a combination of some basic properties of convex functions and Bregman divergences, and it can be found in many OMD textbooks and literatures, e.g., [13] and Chapter 28 of [14]. For completeness, we also present a proof here.

Proof of Lemma 1. Since Ψ is Legendre, $\nabla\Psi$ will explode on $\partial\mathbb{R}_+^K$, which guarantees that the minimizer x' in the definition of $\tilde{z}_t$ in (4) will lie in $int(\mathbb{R}_+^K)$ and $\frac{\partial}{\partial x'}[\langle l_t, x'\rangle + \sigma_t D_\Psi(x', x_t)] = 0$. The bijection property in Lemma 7 then asserts this $\arg\min$ definition is equivalent to the definition in (5) using mirror maps $\nabla\Psi$ and $\nabla\Psi^*$. Since $\bar{\Psi}$ is a Legendre function on $\Delta^{[M,K-1]}$, a similar argument suggests that the definitions for z_t in (3) and (5) are equivalent.

The definition of $\tilde{z}_t$ in (5) implies $l_t = \sigma_t(\nabla\Psi(x_t) - \nabla\Psi(\tilde{z}_t))$. The first order optimality condition of z_t in (3) implies that $\langle \frac{1}{\sigma_t} l_t + \nabla\Psi(z_t) - \nabla\Psi(x_t), y - z_t\rangle \geq 0$ for any $y \in \Delta^{[M,K-1]}$. Hence we have

$$\begin{aligned}
&\langle l_t, x_t - y\rangle \\
&= \langle l_t, x_t - z_t\rangle + \langle l_t, z_t - y\rangle \\
&\leq \sigma_t \langle \nabla\Psi(x_t) - \nabla\Psi(\tilde{z}_t), x_t - z_t\rangle
\end{aligned}$$

$$
\begin{aligned}
&\quad + \sigma_t\langle \nabla\Psi(z_t) - \nabla\Psi(x_t), y - z_t\rangle \\
&\overset{(a)}{=} \sigma_t(D_\Psi(z_t, x_t) + D_\Psi(x_t, \tilde{z}_t) - D_\Psi(z_t, \tilde{z}_t)) \\
&\quad - \sigma_t(D_\Psi(y, z_t) + D_\Psi(z_t, x_t) - D_\Psi(y, x_t)) \\
&= \sigma_t D_\Psi(y, x_t) - \sigma_t D_\Psi(y, z_t) + \sigma_t D_\Psi(x, \tilde{z}_t) - \sigma_t D_\Psi(z_t, \tilde{z}_t) \\
&\le \sigma_t D_\Psi(y, x_t) - \sigma_t D_\Psi(y, z_t) + \sigma_t D_\Psi(x_t, \tilde{z}_t)
\end{aligned}
$$

where (a) uses the following "three-point identity" of Bregman divergences:

$$
D_\Psi(a, b) + D_\Psi(b, c) - D_\Psi(a, c) = \langle \nabla\Psi(c) - \nabla\Psi(b), a - b\rangle.
$$

□

B. Proof of Lemma 4 and Theorem 5

Lemma 4 is Lemma 2 of [1], we adapt the result to M-set semi-bandit and provide a formal proof below.

Proof of Lemma 4. Let $\tilde{x} = \nabla\Psi^*(\sum_{i=1}^h \frac{\sigma_i}{\sigma}\nabla\Psi(z_i))$, we have

$$
\begin{aligned}
\sigma D_\Psi(y, x) &\overset{(a)}{\le} D_\Psi(y, \tilde{x}) \\
&\overset{(b)}{=} \sigma D_{\Psi^*}(\nabla\Psi(\tilde{x}), \nabla\Psi(y)) \\
&= \sigma D_{\Psi^*}(\sum_{i=1}^h \frac{\sigma_i}{\sigma}\nabla\Psi(z_i), \nabla\Psi(y)) \\
&\overset{(c)}{\le} \sigma \cdot \sum_{i=1}^h \frac{\sigma_i}{\sigma} D_{\Psi^*}(\nabla\Psi(z_i), \nabla\Psi(y)) \\
&\overset{(d)}{=} \sum_{i=1}^h \sigma_i D_\Psi(y, z_i)
\end{aligned}
$$

where (a) is due to the Pythagorean theorem for Bregman divergences ($D_\Psi(y, \tilde{x}) = D_\Psi(y, x) + D_\Psi(x, \tilde{x}) \ge D_\Psi(y, x)$), (b) is due to the duality property of Bregman divergences, (c) is due to the convexity of the first argument of Bregman divergences, and (d) uses again the duality property. □

The general regret bound for `Banker-OMD-M` (Theorem 5) can now be obtained by a summation over the single-step regret bounds in Lemma 4 (see [1] Appendix A for a formal inductive proof).

C. Technical Lemmas for Theorem 6

We have the following Lemma 8 stating that we can universally bound $D_\Psi(y, x_0)$ for all $y \in \mathcal{A}$ and $\mathbb{E}[\sigma_t D_\Psi(x_t, \tilde{z}_t) \mid \mathcal{F}_{t-1}]$ for all $t \ge 1$.

Lemma 8. *For $\Psi(x) = -2\sum_{i=1}^K \sqrt{x_i}$ and $x_0 = (M/K, \ldots, M/K)$, we have*

$$
D_\Psi(y, x_0) \le 2\sqrt{MK}
$$

for any $y \in \mathcal{A}$.

In Algorithm 3, for any $t \ge 1$, we have

$$
\mathbb{E}[\sigma_t D_\Psi(x_t, \tilde{z}_t) \mid \mathcal{F}_{t-1}] \le \frac{\sqrt{MK}}{\sigma_t}
$$

where $\tilde{z}_t = \nabla\Psi^*(\nabla\Psi(x_t) - \frac{1}{\sigma_t}\tilde{l}_t)$.

Remark. This upper-bound is $\sqrt{M}$ times larger than the classical MAB setting (i.e., $M = 1$), and the argument need is also similar. To prove Lemma 8, we adapt the proof from Lemma 11 of [1] to fit our feasible action set $\Delta^{[M,K-1]}$ and the choice $x_0 = (M/K, \ldots, M/K)$.

Proof of Lemma 8. The first bound is quite immediate, for $x_0 = (M/K, \ldots, M/K)$, we have

$$
\begin{aligned}
D_\Psi(y, x_0) &= \Psi(y) - \Psi(x_0) - \langle \nabla\Psi(x_0), y - x_0\rangle \\
&= \Psi(y) - \Psi(x_0) \\
&\le 2\sum_{i=1}^K \sqrt{x_{0,i}} \\
&= 2\sqrt{MK}
\end{aligned}
$$

for any $y \in \Delta^{[M,K-1]}$.

For the second bound, in fact, for any choice of Ψ and $t \ge 1$ we have

$$
\begin{aligned}
\sigma_t D_\Psi(x_t, \tilde{z}_t) &= \sigma_t D_{\Psi^*}(\nabla\Psi(\tilde{z}_t), \nabla\Psi(x_t)) \\
&= \sigma_t D_{\Psi^*}(\nabla\Psi(x_t) - \frac{\tilde{l}_t}{\sigma_t}, \nabla\Psi(x_t)) \\
&= \Psi^*(\nabla\Psi(x_t) - \frac{\tilde{l}_t}{\sigma_t}) \\
&\quad - \Psi^*(\nabla\Psi(x_t)) - \langle x_t, -\frac{\tilde{l}_t}{\sigma_t}\rangle \\
&= \frac{\|\tilde{l}_t\|^2_{\nabla^2\Psi^*(\theta_t)}}{2\sigma_t}, \qquad (8)
\end{aligned}
$$

where in the last step we write the Bregman divergence into a second order Lagrange remainder, θ_t is some element inside the line segment connecting $\nabla\Psi(x_t) - \frac{1}{\sigma_t}\tilde{l}_t$ and $\nabla\Psi(x_t)$.

Note that under this particular $\Psi(x) = -2\sum_{i=1}^K \sqrt{x_i}$, we have

- $\nabla^2\Psi^*(\cdot)$ is diagonal,
- The diagonal elements of $\nabla^2\Psi^*(\cdot)$ are non-decreasing on the line segment $[\nabla\Psi(x_t) - \frac{1}{\sigma_t}\tilde{l}_t, \nabla\Psi(x_t)]$,

and we can further upper-bound (8) by $\frac{1}{2\sigma_t}\|\tilde{l}_t\|^2_{\nabla^2\Psi^*(\nabla\Psi(x_t))} = \frac{1}{2\sigma_t}\|\tilde{l}_t\|^2_{\nabla^2\Psi(x_t)^{-1}}$. Therefore,

$$
\begin{aligned}
&\quad \mathbb{E}[\sigma_t D_\Psi(x_t, \tilde{z}_t) \mid \mathcal{F}_{t-1}] \\
&\le \mathbb{E}[\frac{1}{2\sigma_t}\|\tilde{l}_t\|^2_{\nabla^2\Psi(x_t)^{-1}} \mid \mathcal{F}_{t-1}] \\
&= \frac{1}{\sigma_t}\sum_{i=1}^K x_{t,i}^{3/2}\mathbb{E}[\tilde{l}^2_{t,i} \mid \mathcal{F}_{t-1}] \\
&= \frac{1}{\sigma_t}\sum_{i=1}^K x_{t,i}^{3/2}\mathbb{E}\left[\frac{l^2_{t,A_t}A_{t,i}}{x^2_{t,i}} \;\middle|\; \mathcal{F}_{t-1}\right] \\
&\le \frac{1}{\sigma_t}\sum_{i=1}^K x_{t,i}^{-1/2}\mathbb{P}[i \in S_t \mid \mathcal{F}_{t-1}] \\
&= \frac{1}{\sigma_t}\sum_{i=1}^K x_{t,i}^{1/2}
\end{aligned}
$$

$$\leq \frac{\sqrt{MK}}{\sigma_t}$$

where the last step is due to Cauchy-Schwatz inequality. □

The second result we need is an upper-bound of the sum of σ_t^{-1} and the final value of B_T, both from [1]:

Lemma 9 (Theorem 6 of [1])**.** *For any time horizon length $T \geq 1$, let $D = \sum_{t=1}^{T} d_t$ denote the total delay during the first T time steps, then we have*

$$\sum_{t=1}^{T} \sigma_t^{-1} \leq O(\sqrt{T} + \sqrt{D \log D}).$$

Furthermore, at the end of the T-th time slot, we have

$$B_T \leq O(\sqrt{T} + \sqrt{D \log D}).$$

To obtain Theorem 6, we plug the bounds in Lemma 8 and Lemma 9 into Theorem 5 and then take expectation.