Multi-slots Online Matching with High Entropy

Xingyu Lu [1] **Qintong Wu** [1] **Wenliang Zhong** [1]

Abstract

Online matching with diversity and fairness pursuit, a common building block in the recommendation and advertising, can be modeled as constrained convex programming with high entropy. While most existing approaches are based on the "single slot" assumption (i.e., assigning one item per iteration), they cannot be directly applied to cases with multiple slots, e.g., stock-aware top-N recommendation and advertising at multiple places. Particularly, the gradient computation and resource allocation are both challenging under this setting due to the absence of a closed-form solution. To overcome these obstacles, we develop a novel algorithm named Online subGradient descent for Multi-slots Allocation (OG-MA). It uses an efficient pooling algorithm to compute closed-form of the gradient then performs a roulette swapping for allocation, yielding a sub-linear regret with linear cost per iteration. Extensive experiments on synthetic and industrial data sets demonstrate that OG-MA is a fast and promising method for multi-slots online matching.

1. Introduction

Online matching plays a crucial role in many real-world applications, such as online advertising with guaranteed contracts (Mehta et al., 2007; Feldman et al., 2009) and goods recommendation under inventory constraints (Talluri & Van Ryzin, 1998; Zhong et al., 2015). Many works have been done to investigate the theoretical properties of this problem, as well as the industrial deployment. Most of them are built on a bipartite graph $G(\mathbb{A}, \mathbb{T}, \mathbb{E})$ with the *single-slot* assumption. Here $\mathbb{A}$ denotes nodes resource to assign, while $\mathbb{T}$ refers to the node of users arriving one by one. And $\mathbb{E}$ is the edge set representing valid assignments. For each node $t \in \mathbb{T}$, a decision-maker takes an action to match one node from t's neighbors in $\mathbb{A}$ to maximize the sum weights of the matched subset edges under the resource constraints on $\mathbb{A}$.

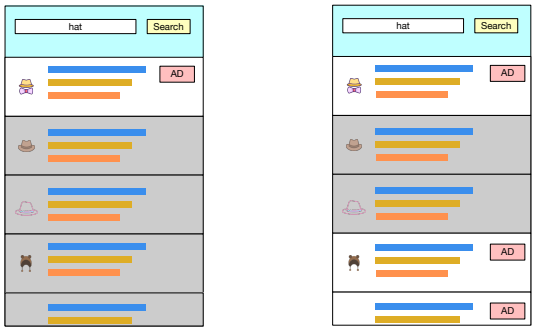

Figure 1: Left figure illustrates the typical online matching application that presents an advertisement in a single slot, and right figure shows a practical search engine results page where several advertisements are blended in multiple slots.

Pioneered works modeled online matching as linear programming (LP) with a primal-dual framework that maintains dual variables induced by resource constraints (Devanur & Hayes, 2009; Agrawal et al., 2014; Li et al., 2020). When a user node arrives, the primal solution is recovered (i.e. picking one item) , and then a gradient descent step is applied to the dual. However, the LP approaches do not take the diversity and fairness into consideration, and thus not suitable for specific tasks. To handle the above issue, the high entropy regularization, encouraging desired properties in advertising and recommendation (Qin & Zhu, 2013; Ahmed et al., 2017; Di Noia et al., 2017), is introduced to online matching (Zhong et al., 2015; Agrawal et al., 2018).

While the above works focus on the *single-slots* setting, there are a large number of applications involving *multi-slots* , such as the product feed recommendation and search result pages with ads (see Figure 1). However, it is non-trivial to extend existing methods to the multi-slots setting with diversity pursuit (Zhang, 2009; Yan et al., 2020). They either rely on a closed-form of gradient computation (Zhong et al., 2015; Agrawal et al., 2018) or assume the primal recovery is computationally lightweight (Balseiro et al., 2020; 2021) , which fail to hold in the presence of $N > 1$ slots. To the best of our knowledge, a high entropy online matching in the multi-slots setting remains unexplored and

[1]Ant Group, Hangzhou, China. Correspondence to: Xingyu Lu <sing.lxy@antgroup.com>, Qintong Wu <qintong.wqt@antgroup.com>, Wenliang Zhong <yice.zwl@antgroup.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

would benefit a broad class of applications.

In this paper, we adopt a random permutation input model. That is, the arriving nodes $\mathbb{T}$ can be picked adversarially at the start, and the arrival order is allowed to be uniformly distributed over all permutations. Consider the problem on a bipartite graph $G_N(\mathbb{A}, \mathbb{T}, \mathbb{E})$ with N-slots of *capacity* $\mathbf{c}$. In detail, c_i is associated to some properties of the i-th slot, e.g. *position bias* (Chen et al., 2020) , and has a big impact on the cumulative reward and resources constraints. In each iteration, the decision-maker only observes the revenue and resource consumption in the hindsight for each node arrival $t \in \mathbb{T}$, and then takes an action $\mathbf{X}_t \in [0,1]^{A\times N}$ to allocate N distinct nodes in $\mathbb{A}$ to N slots. Our goal is to propose a simple and efficient algorithm that generates maximum matching with high entropy and attains sub-linear regret.

1.1. Our Contribution

We develop a novel algorithm, Online subGradient descent for Multi-slots Allocation (OG-MA), to maximize a concave objective consisting of cumulative revenues and high entropy regularizer. In detail, the proposed algorithm first employs the primal-dual framework to update dual variables with online gradient descent(OGD) (Shalev-Shwartz et al., 2011). Next, the proposed algorithm decomposes the recovery of primal solution into a two-step approach. In the first step, we develop a fast Efficiency Pooling Projection (EPP) algorithm to compute gradients of dual variables. In the second step, we propose a Roulette Swapping Allocation (RSA) algorithm to generate the allocation $\mathbf{X}_t \in [0,1]^{A\times N}$.

The proposed OG-MA is simple and efficient for large-scale applications. For each arriving request, our proposed two-steps approach presents an optimal allocation with linear complexity w.r.t. the number of slots N. Meanwhile, the update rule of dual variables is computationally lightweight, which only requires initial dual variables and step size as inputs. We show the effectiveness of the two-steps approach in Proposition 3.1 and prove the optimality of the EPP method in Theorem 5.1.

Under the assumption of the random permutation model, we derive the upper bound for the regret analysis of the proposed algorithm. In detail, our algorithm achieves $\mathcal{O}((\sqrt{K}+\log T)\sqrt{T})$ regret, where T is the number of arriving requests and K is the number of resources constraints. In Theorem 5.3, we show that our regret analysis is valid for general concave revenue functions. When the input model degenerates to a stochastic input model with request i.i.d. drawn, the proposed algorithm still attains $\mathcal{O}(\sqrt{KT})$ regret, which is at the optimal order of existing results.

Extensive experiments on synthetic and industrial data sets validate our computation complexity analysis and sub-linear regret results. Further, we show that OG-MA algorithm achieves positive diversity uplift by the trade-off between the entropy regularizer and the matching revenue.

2. Related Work

The online matching problem has been studied extensively in both theoretical analyses as well as practical implementations. The comparison between our work and the existing literature is summarized in Table 1.

Previous works mainly consider linear revenue functions in the online matching problem. Two prevalent models are adopted: the stochastic input model and the random permutation model. The stochastic input model, a special case of the random permutation model, assumes an online request to be drawn i.i.d. from an unknown distribution. (Devanur et al., 2011) presented an online algorithm with the knowledge of the value of benchmark to attain $\mathcal{O}(\sqrt{T})$ regret, and (Arlotto & Gurvich, 2019) further confirmed that $\mathcal{O}(\sqrt{T})$ is the lowest attainable regret under such setting. (Li & Ye, 2019) improved the regret order to $\mathcal{O}(\log T)$ by periodically solving a linear program with the strongly convex assumption. Compared with the stochastic input model, the random permutation model in our setting is more general to match practical applications. (Devanur & Hayes, 2009; Feldman et al., 2010) presented a similar two-phase dual training algorithm which attains $\mathcal{O}(T^{2/3})$ regret. (Agrawal et al., 2014; Kesselheim et al., 2014) developed a dual-based and a primal-beats-dual-based algorithm, respectively. Although their methods achieved $\mathcal{O}(\sqrt{T})$ regret, an additional linear program is required to solve periodically. Our proposed algorithm only requires a single pass through the input data with straightforward update rules and still enjoys linear complexity. (Li et al., 2020) studied a simple and fast algorithm that attains $\mathcal{O}(\log T\sqrt{T}))$ regret order. Compared to (Li et al., 2020), our results further improve the regret dependency of the number of resources K from $\mathcal{O}(K)$ to $\mathcal{O}(\sqrt{K})$, and work for general concave revenue functions.

Research interests of high entropy matching mainly focus on the single-slot setting. (Zhong et al., 2015) developed a fractional allocation using the dual-descent algorithm in stock constrained recommendation. (Agrawal et al., 2018) provided an offline multi-rounds proportional matching algorithm to maintain the diversity. The sub-problem of recovering primal solution has corresponding closed-formed solution in the aforementioned literature, but the desired properties cannot be extended to the multi-slots setting directly. This paper extends this line of research to the multi-slots setting and presents a fast and efficient online algorithm.

The high entropy matching also extends to a line of works on online allocation problems with concave revenues. Previous works studied general concave maximum objectives with convex feasibility constraints. (Agrawal & Devanur,

Table 1: Comparsion of our work with the existing literature on online matching problems

Papers	Objective	Input model[1]	Violation risk	Linear complexity	Regret
(Devanur & Hayes, 2009; Feldman et al., 2010)	Linear	R.P.	No	No	$\mathcal{O}(T^{2/3})$
(Agrawal et al., 2014; Kesselheim et al., 2014)					$\mathcal{O}(\sqrt{T})$
(Li et al., 2020)	Linear	R.P.	Yes	Yes	$\mathcal{O}(\log T\sqrt{T})$
(Devanur et al., 2011)	Linear	I.I.D.	No	No	$\mathcal{O}(\sqrt{T})$
(Li & Ye, 2019)					$\mathcal{O}(\log T)$
(Agrawal & Devanur, 2014)	Concave	R.P.	Yes	No	$\mathcal{O}(\sqrt{T})$
(Balseiro et al., 2020; 2021)	Concave	I.I.D.	No	Yes	$\mathcal{O}(\sqrt{T})$
Our paper	Concave	R.P.	No	Yes	$\mathcal{O}(\log T\sqrt{T})$

[1] R.P. denotes the random permutation model and I.I.D. denotes the stochastic input model with samples i.i.d. drawn.

2014) adopted the random permutation model and developed a fast dual-decent online algorithm when taking the knowledge of the value benchmark. Although their results attain better regret order than ours, their algorithm requires a complex step to periodically solve a convex optimization program if the benchmark is unknown. In addition, their work allows the violation of resource constraints, which is hardly acceptable in real-world applications. Instead of the random permutation model, (Balseiro et al., 2020) solved similar problems using a dual mirror descent algorithm with the assumption of i.i.d. input model. We extend the regret analysis to the random permutation model. When the input model degenerates to the stochastic i.i.d. model, our analysis admits the same regret $\mathcal{O}(\sqrt{T})$ but is more concise from a sub-gradient descent perspective. (Balseiro et al., 2021) studied a class of regularized online allocation problems that explicitly considers fairness among the resource consumption. However, their algorithms cannot extend to handle the diversity within each request in the multi-slots setting. Last but not least, the aforementioned researches all assumed that the recovery of the primal solution is computationally lightweight under the primal-dual framework, which is often contrary in real-world applications.

3. Problem Formulation

For the purpose of description, we adopt the terminology from online advertising, where advertisements and users represent the two sides of a bipartite graph $G_N(\mathbb{A}, \mathbb{T}, \mathbb{E})$.

First, we define $\mathbb{T}$ as a set of arriving users and $\mathbb{A}$ as the advertisements. The candidate advertisements for any $t \in \mathbb{T}$ can be represented by a subset $\mathbf{E}_t$ of edge set $\mathbb{E} = \{(t, a) : t \in \mathbb{T}, a \in \mathbf{E}_t\}$ with size A_t. Next, we define the decision matrix $\mathbf{X}_t \in [0, 1]^{A\times N}$ that allocates N advertisements to N slots for t-th user, where each entry $x_{t,a,n}$ is indexed by an advertisement a and a slot n. Then, the following conditions must be met during a user's visit:

- Each slot must be assigned with an advertisement.
- An advertisement can be allocated to at most one slot.

The above conditions can be transformed to following constraints:

$$\mathcal{X} = \begin{bmatrix} \sum_{a\in\mathbf{E}_t} x_{t,a,n} & = 1, \forall t \in \mathbb{T}, 1 \le n \le N \\ \sum_{1\le n\le N} x_{t,a,n} & \le 1, \forall t \in \mathbb{T}, a \in \mathbf{E}_t \end{bmatrix} \tag{1}$$

To characterize the capacity of impressions among different slots, we introduce vector $\mathbf{c} = (c_1, c_2, \ldots, c_N) \in (0, 1]^N$ to incorporate the N-slots information, where c_n denotes the impressions capacity for the slot n. To simplify the description, the slots capacity vector $\mathbf{c}$ is assumed to be the same for all arriving users and satisfies the following non-increasing condition without losing generality:

$$1 \ge c_1 \ge c_2 \ge \cdots \ge c_N > 0$$

We construct the following multi-slots matching problem with slots' impressions capacity, resource constraints, and a high entropy regularizer:

$$\begin{aligned} &\max_{\mathbf{X}\in\mathcal{X}} \sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{X}_t\mathbf{c} + \alpha\mathcal{H}(\mathbf{X}_t\mathbf{c}) \\ &s.t. \sum_{t=1}^{T} \mathbf{M}_t^\top \mathbf{X}_t\mathbf{c} \le T\mathbf{B} \end{aligned} \tag{2}$$

where $\mathbf{r}_t \in \mathbb{R}_+^A$ is the revenue vector of user t's candidate advertisements with the elements setting to zero if $a \notin \mathbf{E}_t$, $\mathbf{M}_t \in \mathbb{R}_{++}^{A\times K}$ is the entry-wise positive cost matrix on K

resources, and $\mathbf{B} \in \mathbb{R}^K_{++}$ is the resources constraint vector. In order to guarantee a feasible solution given the number of slots N for each $t \in \mathbb{T}$, we assume that there exists at least N advertisements with revenue entries $r_{t,a} = 0$ and resource consumption entries $m_{t,a,k} = 0$. In addition, the revenue $\mathbf{r}_t$ and resource consumption $\mathbf{M}_t$ of the matching problem in (2) are all scaled by the advertisements' allocated impressions $\mathbf{X}_t\mathbf{c}$ for each $t \in \mathbb{T}$. The diversity of the matching is introduced by an entropy regularizer $\mathcal{H} \in \mathbb{R}^A \to \mathbb{R}$ defined as:

$$\mathcal{H}(\mathbf{x}) = -\sum_{a \in \mathbb{A}} x_a \log(x_a) \tag{3}$$

In equation (2), α denotes a parameter to control the trade-off between revenue and diversity.

In the online setting, we adopt the random permutation model (further discussed in Assumption 5.2). In each time t, the algorithm receives a request $(\mathbf{r}_t, \mathbf{M}_t)$ and estimates a fractional matching $\hat{\mathbf{X}}_t$. The cumulative revenue is defined under the random permutations σ over $1, 2, \ldots, T$:

$$R(\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \ldots, \hat{\mathbf{X}}_T|\sigma) = \mathbb{E}_\sigma[\sum_{t=1}^{T} \mathbf{r}_t^\top \hat{\mathbf{X}}_t\mathbf{c} + \alpha\mathcal{H}(\hat{\mathbf{X}}_t\mathbf{c})]$$

The remaining resources are considered in the way that $\sum_{s=1}^{t-1} \mathbf{M}_s^\top \widetilde{\mathbf{X}}_s\mathbf{c} + \mathbf{M}_t^\top \hat{\mathbf{X}}_t\mathbf{c} \leq T\mathbf{B}$ must be satisfied when generating a fractional matching $\hat{\mathbf{X}}_t$. Here, $\widetilde{\mathbf{X}_t} \in \{0,1\}^{A \times N}$ is a realization of $\hat{\mathbf{X}}_t$ by the decision-maker $\mathcal{A}$. Denote θ as the random variable that determines the realization of decision-maker $\mathcal{A}$, the expect revenue of $\mathcal{A}$ can be represented by:

$$R(\mathcal{A}|\sigma) = \mathbb{E}_\theta[R(\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \ldots, \hat{\mathbf{X}}_T|\sigma)]$$

When all parameters of T users (i.e. $\{(\mathbf{r}_t, \mathbf{M}_t)\}_{t=1}^T$) are known in advance, we can solve the optimal matching and obtain the optimal revenue:

$$R^* = \max_{\mathbf{X} \in \mathcal{X}} \left\{ \begin{array}{l} \sum_{t=1}^T \mathbf{r}_t^\top \mathbf{X}_t\mathbf{c} + \alpha\mathcal{H}(\mathbf{X}_t\mathbf{c}), \\ s.t. \sum_{t=1}^T \mathbf{M}_t^\top \mathbf{X}_t\mathbf{c} \leq T\mathbf{B} \end{array} \right\} \tag{4}$$

The regret of the decision-maker is defined as the following:

$$\text{Regret}(\mathcal{A}|\sigma) = R^* - R(\mathcal{A}|\sigma) \tag{5}$$

Our goal is to design an algorithm that efficiently solves the multi-slots matching problem (2) and constructs a fast real-time allocation. Meanwhile, low regret must be attained without violating the intended constraints.

3.1. The Dual Problem

Our main algorithm (Algorithm 1 in Section 4) is derived upon the dual-descent algorithm. The primary challenge lies in the complexity of solving a sub-problem in dual formation of (2). Given dual variables, the recovery of $\mathbf{X}_t$ requires $\mathcal{O}(N^3 A_t^3)$ complexity in the worst case where $\mathcal{O}(NA_t)$ is the number of variables in $\mathbf{X}_t$. We address the challenge by introducing an intermediate variable $\mathbf{y}_t := \mathbf{X}_t\mathbf{c}$ to decompose the sub-problem into a two-step optimization. In step one, we develop a formation of $\mathbf{y}_t$ in which the number of variables is reduced to $\mathcal{O}(A_t)$. In step two, an real-time allocation $\mathbf{X}_t$ is recovered given $\mathbf{y}_t$.

Step-One (*Optimization of* $\mathbf{y}_t$). The $\mathbf{y}_t$ characterizes advertisements' *expected impressions* in all slots given $\mathbf{X}_t$. Then, we can transform the problem (2) as following:

$$\begin{aligned} &\max_{\mathbf{y} \in \mathcal{Y}} \sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{y}_t + \alpha\mathcal{H}(\mathbf{y}_t) \\ &s.t. \sum_{t=1}^{T} \mathbf{M}_t^\top \mathbf{y}_t \leq T\mathbf{B} \end{aligned} \tag{6}$$

where the domain constraints $\mathcal{Y}$ is defined as:

$$\mathcal{Y} = \begin{bmatrix} \sup_{\mathbf{E}_t(n)} \sum_{a \in \mathbf{E}_t(n)} y_{t,a} & \leq \sum_{s=1}^n c_s, \forall t \in \mathbb{T}, n < N \\ \sum_{a \in \mathbf{E_t}} y_{t,a} & = \sum_{n=1}^N c_n, \forall t \in \mathbb{T} \end{bmatrix}$$

and $\mathbf{E}_t(n)$ represents a subset of advertisements $\mathbf{E}_t$ with size n. The constraints $\mathcal{Y}$ implies that cumulative impressions of the top-n advertisements' cannot exceed the top-n slots' impressions capacity within a multi-slots allocation.

Step-Two (*Recovery of* $\mathbf{X}_t$). A fractional matching action $\mathbf{X}_t$ is recovered with an estimated $\hat{\mathbf{y}}_t$. The recovery procedure is characterized by a linear system:

$$\hat{\mathbf{X}}_t = \arg_{\mathbf{X}_t \in \mathcal{X}} \{\mathbf{X}_t\mathbf{c} = \hat{\mathbf{y}}_t\} \tag{7}$$

Proposition 3.1. *The optimal* $\mathbf{X}_t^*$ *of problem* (2) *is equivalent to the* $\hat{\mathbf{X}}_t$ *solved by the two-step problem* (6) *and* (7)*.*

Next, we introduce dual variables $\boldsymbol{\lambda} \in \mathbb{R}_+^K$ of the resources constraints to obtain the Lagrangian dual of (6) in a max-min form. Since the primal objective is concave, we can swap the min and max under the strong duality as following:

$$\begin{aligned} \min_{\boldsymbol{\lambda} \geq 0} \max_{\mathbf{y} \in \mathcal{Y}} L(\boldsymbol{\lambda}, \mathbf{y}) = &\sum_{t=1}^{T} [\mathbf{r}_t^\top \mathbf{y}_t + \alpha\mathcal{H}(\mathbf{y}_t)] \\ &+ \boldsymbol{\lambda}^\top (T\mathbf{B} - \sum_{t=1}^{T} \mathbf{M}_t^\top \mathbf{y}_t) \end{aligned} \tag{8}$$

3.2. Properties of the Optimal Primal Solution of (8)

We present the analysis of (8) to elaborate an efficient algorithm. Given $\boldsymbol{\lambda}$, we define $v_{t,a}$ as the *contribution* value to the maximum objective of advertisement a for user t, and let $e_{t,a}$ be the *efficiency* value for primal solution $y_{t,a}$ by:

$$v_{t,a} := \exp(\frac{r_{t,a} - \sum_k \lambda_k m_{t,a,k}}{\alpha});\ e_{t,a} := \frac{v_{t,a}}{y_{t,a}}.$$

Next, we arrange the advertisements index $\mathbf{E}_t$ with size A_t in the order:

$$v_{t,1} \geq v_{t,2} \geq \cdots \geq v_{t,A_t}, \forall t \in \mathbb{T}. \tag{9}$$

The following proposition shows that the optimal $y^*_{t,a}$ and its efficiency value $e^*_{t,a}$ are in the same order as $v_{t,a}$.

Proposition 3.2. *For the optimal solution $\mathbf{y}^*_t(\boldsymbol{\lambda})$ of (8) given $\boldsymbol{\lambda}$, it holds for all t that:*

$$y^*_{t,1} \geq y^*_{t,2} \geq \cdots \geq y^*_{t,A_t}; e^*_{t,1} \geq e^*_{t,2} \geq \cdots \geq e^*_{t,A_t}.$$

Then, we can divide the $\mathbf{E}_t$ into blocks that share the same efficiency value. The following gives a formal definition.

Definition 3.3. For any feasible primal solution $\mathbf{y}_t$ of (8), there exists a unique disjoint block set $\{\mathbb{B}_r\}_{r=1}^{l}$ that can divide $\mathbf{E}_t$ (sorted by (9)) into l blocks with:

1. $e_{t,i} = e_{t,j}, \forall i, j \in \mathbb{B}_r, 1 \leq r \leq l$;

2. $e_{t,i} \neq e_{t,j}, \forall i \in \mathbb{B}_r, j \in \mathbb{B}_{r+1}, 1 \leq r \leq l-1$.

The efficiency value of block $\mathbb{B}_r$ is defined as :

$$E(\mathbb{B}_r) = \frac{\sum_{a \in \mathbb{B}_r} v_{t,a}}{\sum_{a \in \mathbb{B}_r} y_{t,a}}. \tag{10}$$

Further, each block $\mathbb{B}_r$ can be formed by its elements:

$$\mathbb{B}_{(p,q]} := \{a | p < a \leq q, a \in \mathbf{E}_t\}$$

Then, we present two propositions to show additional properties of optimal solution $\mathbf{y}^*_t(\boldsymbol{\lambda})$ which help to design the algorithm. Proofs are shown in Appendix C and D.

Proposition 3.4. *(Equal efficiency within a block). $\{\mathbb{B}_r\}^*$ is the block set of $\mathbf{y}^*_t(\boldsymbol{\lambda})$, it holds for any block $\mathbb{B}_r := \mathbb{B}_{(p,q]} \in \{\mathbb{B}_r\}^*$:*

$$E(\mathbb{B}^*_{(p,r]}) = E(\mathbb{B}^*_{(p,q]}), \forall p < r < q.$$

Proposition 3.5. *(Decreasing efficiency between blocks). $\{\mathbb{B}_r\}^*$ is the block set of $\mathbf{y}^*_t(\boldsymbol{\lambda})$, if $y_{t,r}$, $y_{t,r+1}$ belong to different blocks, the following inequality holds:*

$$E(\mathbb{B}^*_{(p,r]}) > E(\mathbb{B}^*_{(r,q]}), \forall p < r < q.$$

Remark 3.6. When slot $N = 1$ and capacity $\mathbf{c} = 1$, $y^*_{t,a}(\boldsymbol{\lambda}) = v_{t,a} / \sum_a v_{t,a}$ is a closed-formed solution of the dual problem (8), and it satisfies Proposition 3.2, 3.4, and 3.5.

Remark 3.7. We introduce the advantage of pool adjacent violators algorithm (De Leeuw et al., 2010), a widely used method in the isotonic regression with a quadratic minimum objective, into our algorithm design. Although the ideas are similar, the objective in our setting is essentially different. In the next section, we propose a fast algorithm through the analysis in Proposition 3.2, 3.4, and 3.5.

Algorithm 1 Online subGradient descent for Multi-slots Allocation (OG-MA)

Input: User set $\mathbb{T}$, the step-size η; and initialize dual variables $\boldsymbol{\lambda}_0 = \mathbf{0}$.
for $t = 1$ **to** T **do**
Receive a stochastic request with $(\mathbf{r}_t, \mathbf{M}_t)$.
Solve the expected impressions $\hat{\mathbf{y}}_t$ for all advertisements using **efficiency pooling projection**;
Update the allocated impressions under the remaining resources:

$$\widetilde{\mathbf{y}}_t = \begin{cases} \hat{\mathbf{y}}_t, & \text{if} \sum_{s=1}^{t-1} \mathbf{M}_s^\top \widetilde{\mathbf{X}}_s \mathbf{c} + \mathbf{M}_t^\top \hat{\mathbf{y}}_t \leq T\mathbf{B}, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{11}$$

Make the realization $\widetilde{\mathbf{X}}_t$ of primal solution $\mathbf{X}_t$ with given $\widetilde{\mathbf{y}}_t$ by **roulette swapping allocation**.
Compute gradients $\mathbf{g}(\boldsymbol{\lambda}_t)$ of $\boldsymbol{\lambda}_t$ where:

$$\mathbf{g}(\boldsymbol{\lambda}_t) := \mathbf{B} - \mathbf{M}_t^\top \hat{\mathbf{y}}_t.$$

Update $\boldsymbol{\lambda}$ by **projected subgradient descent**:

$$\boldsymbol{\lambda}_{t+1} = \text{Proj}_{\boldsymbol{\lambda} \geq 0}\{\boldsymbol{\lambda}_t - \eta \mathbf{g}(\boldsymbol{\lambda}_t)\} \tag{12}$$

end for

4. Multi-Slots Online Algorithm

The proposed OG-MA algorithm is summarized in Algorithm 1 consisting of three steps. First, the Efficiency Pooling Projection (EPP) method is proposed to solve an estimated $\hat{\mathbf{y}}_t(\boldsymbol{\lambda})$ of the dual problem (8). The allocated impressions $\widetilde{\mathbf{y}}_t$ of advertisements is updated by (11) given quantities of remaining resources. Next, a Roulette Swapping Allocation (RSA) algorithm is adopted to solve step-two problem (7) and make a realization $\widetilde{\mathbf{X}}_t$. At last, since the dual (8) is convex with respect to $\boldsymbol{\lambda}$, OGD is employed to update the dual variables $\boldsymbol{\lambda}$ with project operation.

Efficiency Pooling Projection (*Given $\boldsymbol{\lambda}$ and obtain $\mathbf{y}_t$*). We assume that the candidate advertisements $\mathbf{E}_t$ have been sorted by $v_{t,1} \geq v_{t,2} \geq \cdots \geq v_{t,A_t}$. In the initialization phase $l = 0$, we allocate the i-th advertisements to the i-th slot, so that we can decompose all advertisements $\mathbf{E}_t$ into $N+1$ blocks $\{\mathbb{B}^{(0)}_r\}_{r=1}^{N+1}$ as:

$$\mathbb{B}^{(0)}_r := \begin{cases} \{r\}, \forall 1 \leq r \leq N; \\ \{a | N < a \leq A_t\}, r = N+1. \end{cases} \tag{13}$$

The efficiency of each block is initialized as:

$$E(\mathbb{B}_r) = \frac{\sum_{a \in \mathbb{B}_r} v_{t,a}}{\sum_{a \in \mathbb{B}_r} c_a \mathbb{I}(1 \leq a \leq N)}. \tag{14}$$

Algorithm 2 shows how to generate an optimal block set $\{\mathbb{B}_r\}$. In each iteration l, if there exists two adjacent blocks

Algorithm 2 Efficiency Pooling Projection (EPP)

Input: User request $(\mathbf{r}_t, \mathbf{M}_t)$, dual variable $\boldsymbol{\lambda}$.
Sort $\mathbf{E}_t$ in decreasing order by $v_{t,a}$.
Initialize $\mathbf{E}_t$ into blocks $\{\mathbb{B}_r^{(0)}\}_{r=1}^{N+1}$ by (13), compute efficiency value $E(\mathbb{B}_r^0)$ by (14) and set $l = 0$.
repeat
  Step1. Merge $\mathbb{B}^{(l)}$-blocks if $E(\mathbb{B}_r^{(l)}) \le E(\mathbb{B}_{r+1}^{(l)})$.
  Step2. Update the merged blocks $\mathbb{B}_r^{(l+1)} := \mathbb{B}_r^{(l)}$ and efficiency value $E(\mathbb{B}_r^{(l+1)})$ for all r, i.e..
  Step3. If exists $E(\mathbb{B}_r^{(l+1)}) \le E(\mathbb{B}_{r+1}^{(l+1)})$, then increase $l = l + 1$ and go back to Step1.
until $E(\mathbb{B}_r^{(l)}) > E(\mathbb{B}_{r+1}^{(l)})$ for all block r.
Output: $\hat{y}_{t,a} = v_{t,a}/E(\mathbb{B}_r), \forall a \in \mathbb{B}_r$ and block index r.

with increasing efficiency $E(\mathbb{B}_r^{(l)}) \le E(\mathbb{B}_{r+1}^{(l)})$, a pooling action is taken to merge blocks $\mathbb{B}_r^{(l)} = \mathbb{B}_r^{(l)} \cup \mathbb{B}_{r+1}^{(l)}$ and to enforce all elements within the new block share the same efficiency by (14). The algorithm stops when all the blocks' efficiency decreases, e.g., $E(\mathbb{B}_r^{(l)}) > E(\mathbb{B}_{r+1}^{(l)})$. At this point, properties in Proposition 3.4 and Proposition 3.5 are properly ensured. Finally, the assigned impressions $\hat{\mathbf{y}}_t$ are obtained with minimal cost by $\hat{y}_{t,a} = v_{t,a}/E(\mathbb{B}_r), \forall a \in \mathbb{B}_r$ in the Definition 3.3.

Roulette Swapping Allocation(*Given* $\mathbf{y}_t$ *and recover* $\mathbf{X}_t$) To achieve the expected impressions $\widetilde{\mathbf{y}}_t$, Algorithm 3 is proposed to allocate advertisements to slots by swapping their positions. Let $r(a)$ be the ranking order and $y_{t,a}$ be the actual allocated impressions of advertisement a. We allocate the i-th advertisement sorted by $v_{t,a}$ to the i-th slot in the initialize step. Then, we have $r(a) = a, \forall 1 \le a \le A_t$ and $y_{t,a} = c_a$ when $a \le N$, otherwise, $y_{t,a} = 0$. In each round, we compare the advertisement j's impressions $y_{t,j}$ with the expected value $\widetilde{y}_{t,j}$, and then we perform the swapping operation on $s \in \mathbb{S}$ with probability p_s computed by (15). This step in Algorithm 3 utilizes the excess impressions of advertisements in index set $\mathbb{S}$ to make up for under-allocated advertisements .

The swapping operation in Algorithm 3 is always feasible. Given $y_{t,s} > \widetilde{y}_{t,s} \ge \widetilde{y}_{t,j}$, the swapping probability p_s and the cumulative value $\sum_{s\in\mathbb{S}} p_s$ both lie in $[0, 1]$.

Projected subGradient Descent (*Online update* $\boldsymbol{\lambda}$). As shown in Algorithm 1, once an allocation $\widetilde{\mathbf{X}}_t$ is recovered, the gradient $\mathbf{g}(\boldsymbol{\lambda}_t)$ of $\boldsymbol{\lambda}_t$ is readily obtained by $\mathbf{g}(\boldsymbol{\lambda}_t) := \mathbf{B} - \mathbf{M}_t^\top \hat{\mathbf{y}}_t$. Then, the descent step of dual variables $\boldsymbol{\lambda}_{t+1}$ is given by (12) with step-size η.

4.1. Time Complexity

The proposed OG-MA enjoys linear complexity w.r.t. the number of slots N. For each request t, the online allocation

Algorithm 3 Roulette Swapping Allocation (RSA)

Input: Expected impressions $\widetilde{\mathbf{y}}_t$ computed by (11).
Initialize position order $r(a) = a$, the expectation of allocated impressions $y_{t,a} = c_a \mathbb{I}(1 \le a \le N), \forall a \in \mathbf{E}_t$ and index set $\mathbb{S} = \{\}$.
for $j = 1$ **to** A_t **do**
  if $y_{t,j} > \widetilde{y}_{t,j}$ **then**
    Put j into the index set: $\mathbb{S} = \mathbb{S} \cup \{j\}$
  else
    Swap $r(j)$ and $r(s), s \in \mathbb{S}$ with probability:

$$p_s = \frac{\widetilde{y}_{t,j} - y_{t,j}}{y_{t,s} - y_{t,j}} \frac{\widetilde{y}_{t,s} - y_{t,s}}{\sum_{s'\in\mathbb{S}}(\widetilde{y}_{t,s'} - y_{t,s'})}. \tag{15}$$

    Update the allocated impressions of j by $y_{t,j} = \widetilde{y}_{t,j}$
    Update the allocated impressions of $s \in \mathbb{S}$ by:

$$y_{t,s} = (1 - p_s) y_{t,s} + p_s y_{t,j},$$

    and then remove s from $\mathbb{S}$ if $y_{t,s} = \widetilde{y}_{t,s}$.
  end if
end for
Output: Allocate $a \in \mathbf{E}_t$ to $r(a)$-th slot if $r(a) \le N$.

starts by sorting $\mathbf{E}_t$, and then applies EPP and RSA. The complexity of sorting $\mathbf{E}_t$ is $\mathcal{O}(A_t \log A_t)$. In EPP method, $N + 1$ blocks are initialized at first and then at least two adjacent blocks are merged in each iteration. The merge operation obtains $\mathcal{O}(1)$ complexity and the number of iterations is at most N, so that the EPP achieves $\mathcal{O}(N)$ linear time complexity. In RSA algorithm, the update rule $y_{t,j} = \widetilde{y}_{t,j}$ in each round j ensures that the swap operation is taken at most A_t times. In each swapping operation, the time cost comes from computing the probabilities $p_s \in \mathbb{S}$ ($\mathbb{S}$ contains at most N elements). Thus, the time complexity of the RSA is $\mathcal{O}(NA_t)$. Combining the above results, the total complexity of OG-MA is $\mathcal{O}(N + NA_t + A_t \log A_t)$ for each t, outperforming the $\mathcal{O}(N^3 A_t^3)$ complexity of directly solving the optimal $\mathbf{X}_t$ by a general solver.

5. Regret Analysis

In this section, we explain the optimality of the solution $\hat{\mathbf{y}}_t$ solved by EPP method, and present the regret bound of OG-MA under the random permutation model.

5.1. The Optimality of EPP

Theorem 5.1. *The block set* $\{\mathbb{B}_r\}$ *and the primal solution* $\hat{\mathbf{y}}_t$ *solved by EPP method is optimal with given* $\boldsymbol{\lambda}$*. Suppose* L_t *is the block size of* $\{\mathbb{B}_r\}$*, the dual problem* (8) *can be formulated as the convex optimization of dual variable* $\boldsymbol{\lambda}$*:*

$$\min_{\boldsymbol{\lambda}\geq 0} L(\boldsymbol{\lambda}) = T\boldsymbol{\lambda}^\top \boldsymbol{B} + \sum_{t=1}^{T}\sum_{r=1}^{L_t} \alpha I(\mathbb{B}_r)(\ln V(\mathbb{B}_r) - \ln I(\mathbb{B}_r))$$

where $I(\mathbb{B}_r) = \sum_{a\in\mathbb{B}_r} \hat{y}_{t,a}$ *and* $V(\mathbb{B}_r) = \sum_{a\in\mathbb{B}_r} v_{t,a}$.

Proof sketch. The proof of Theorem 5.1 consists of two parts. First, we apply the results in Proposition 3.2 and further present the existence of an optimal $\{\mathbb{B}_r\}^*$ to attain the results in Theorem (5.1). Then, we use the previous two propositions to show that the block set generated by Algorithm 2 is optimal in that it can neither be divided (conflicted with Proposition 3.5) nor merged (conflicted with Proposition 3.4) further. Complete proof is shown in Appendix E.

5.2. Regret Bound

To derive regret bound, we adopt the random permutation model. In this model, arriving users $t \in \mathbb{T}$ with $\{(\mathbf{r}_t, \mathbf{M}_t)\}_{t=1}^T$ can be picked adversarially at the start, and the arrival order of each $(\mathbf{r}_t, \mathbf{M}_t)$ is uniformly distributed over all the permutations. The random permutation model is more general than the stochastic input model in which users arrive i.i.d. from an unknown distribution.

We state the following assumptions to formalize the random permutation model in the multi-slots setting:

Assumption 5.2. (Assumption on random permutation.) We assume that:

(i). The request $(\mathbf{r}_t, \mathbf{M}_t)$ arrives in a random order σ.

(ii). For each request $(\mathbf{r}_t, \mathbf{M}_t)$, there exits $\overline{r} \in \mathbb{R}_{++}$ and $\overline{m} \in \mathbb{R}_{++}$ such that $||\mathbf{r}_t||_\infty \leq \overline{r}$ and $||\mathbf{M}_t^\top \mathbf{X}_t \mathbf{c}||_\infty \leq \overline{m}$.

(iii). There exists $\overline{b}$, $\underline{b}$ such that $\underline{b} \leq B_k \leq \overline{b}, \forall 1 \leq k \leq K$.

Assumption (i) states the permuted observation of the input requests. Assumptions (ii) and (iii) provide a bound of the input request and resources constraints respectively. In the next theorem, $\overline{r}$, $\overline{m}$, $\overline{b}$, and $\underline{b}$ will appear in the analysis of regret bound.

Theorem 5.3. *Consider the OG-MA with step-size* $\eta > 0$ *and initial dual solution* $\boldsymbol{\lambda}_0 = 0$. *Suppose Assumption 5.2 is satisfied, the regret can be upper bounded by:*

$$\begin{aligned} \textit{Regret}(\mathcal{A}|\sigma) \leq & \frac{K(\overline{m}^2 + \overline{b}^2)}{2}\eta T + \frac{C^2}{\underline{b}^2 \eta} + KC\log T \\ & + 2\sqrt{2}\overline{m}\frac{C}{\underline{b}}\log T\sqrt{T} + (K + 2\frac{\overline{m}}{\underline{b}})C \end{aligned}$$

where $C = (\sum_{n=1}^N c_n)(\overline{r} + \alpha \log A)$ *and* A *is the number of advertisements.*

Proof sketch. We prove the theorem in three steps. First, although our algorithm can not exceed the resource constraints in decision-making, we prove that the loss of primal objective is well bounded by the analysis technique of the stopping time (Balseiro et al., 2020). In other words, resources will not be depleted too early. Second, we consider the cumulative revenue without the concern of resources violation. Given $\boldsymbol{\lambda}_t$ in each arrival t, there exists an objective gap between the coming unobserved requests and the total requests in the random permutation model. We present two lemmas F.3 and F.4 to further bound the primal performance of our algorithm. Finally, we put the above two steps together to obtain the result of the theorem. All proofs are shown in Appendix F.

When the step-size η is chosen by $\mathcal{O}(1/\sqrt{KT})$ in Theorem 5.3, we show that our algorithm attains a $\mathcal{O}((\sqrt{K} + \log T)\sqrt{T})$ regret, and therefore our proposed OG-MA achieves sub-linear regret as the number of arrivals and resource constraints vary. Although the algorithms (Agrawal et al., 2014; Kesselheim et al., 2014; Agrawal & Devanur, 2014) attain better regret than ours, they require known estimation on the benchmark or costly computation of periodically solving large-scale optimization problems. Adopting the stochastic input model, we can obtain the same $\mathcal{O}(\sqrt{KT})$ regret as (Balseiro et al., 2020) with a more concise analysis by eliding the Lemma F.3. Moreover, the regret analysis in high entropy matching can extend to general matching with a concave objective.

Remark 5.4. By choosing step-size $\eta = \mathcal{O}(C/\sqrt{KT})$, the regret order is $\mathcal{O}(C(\sqrt{K} + \log T)\sqrt{T})$. The C in Theorem 5.3 represents an upper bound to the maximum primal objective of an allocation. It is related to the top-N advertisements' revenue and the total impressions capacity $\sum_{n=1}^N c_n$. Adopting the position-based click model (Craswell et al., 2008) with $c_n = 1/n^\gamma$, which states the fact that users generally have a higher probability of focusing on the top slots over the bottom slots, we imply that our regret is sub-linear w.r.t. N:

- When $\gamma = 1$, our regret is of order $\mathcal{O}(\log N)$;
- When $\gamma = \frac{1}{2}$, our regret is of order $\mathcal{O}(\sqrt{N})$.

6. Experiments

In this section, we elaborately design the numeric experiment to reveal the difficulty of directly solving the primal problem when the number of slots increases. Next, we consider a large-scale problem, namely display ads, that aims at maximizing total revenue under budget constraints in the search system. The experiment results verify our regret analysis and confirm the diversity uplift by incorporating the high entropy regularizer.

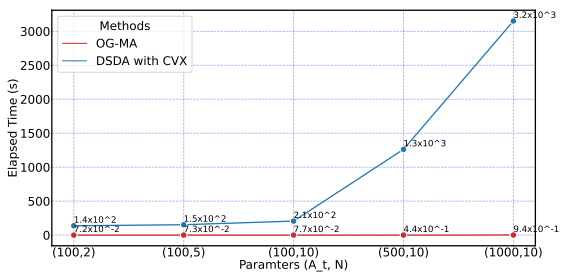

Figure 2: Computing time required escalating the scale of the problem by tweaking N and A_t.

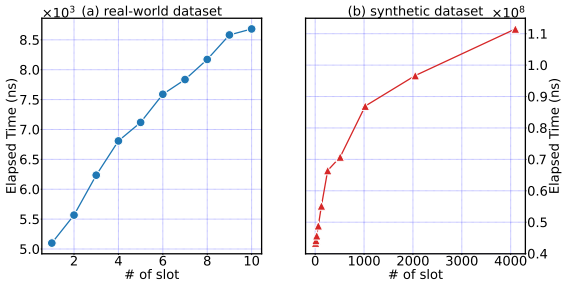

Figure 3: Elapsed inference time for different N.

6.1. Experimental Setup

Synthetic Dataset. We generate the synthetic data to simulate the request with $(\mathbf{r}_t, \mathbf{M}_t)$ at time $t = 1, 2, \ldots, T$. In detail, we sample the $\mathbf{r}_t$ following the approach of (Zhong et al., 2015) and set the amount of resource consumption to the constant $m_{t,a,k} = 1$ for each a and k. Detailed data pre-processing procedure can be found in Appendix G.1.

Industrial Dataset. We utilize *AliExpress Searching System Dataset - France* to evaluate the performance of the proposed algorithm on a large-scale problem. The dataset contains 570,288 users and 8,822,801 samples in total. Each user t has a corresponding revenue function $r_{t,a} = f(t, a)$ for evaluating a list of advertisements in $\mathbb{A}$. Next, we artificially construct 20 budget constraints $\mathbf{B}$ in total and limit $N = 10$ slots for each result page where advertisements are expected to be allocated. The k-th constraint value is defined as following: $b_k = C * T/\sqrt{A}$, where C is the total impression capacity $C = \sum_i^n c_i$, T and A denote the number of users and ads, respectively. The amount of resource consumption is set to the constant $m_{t,a,k} = 1$ for each advertisement. The description of estimated revenue function $f(t, a)$ can be found in Appendix G.2.

Experiment Protocol. We construct the user set $\mathbb{T}$ by randomly sampling from the dataset, and the size of $\mathbb{T}$ is set to $T = 10000$. Next, we randomly permute the set $\mathbb{T}$ to generate the random permutation σ. In the online setting, users arrive sequentially following the order of σ with the estimated revenue vector $\mathbf{r_t}$ and a global resource matrix $\mathbf{M}_t$. And the decision-maker is expected to solve the multi-slots matching with full knowledge of the capacity $\mathbf{c}$. The choice of the impression capacity $\mathbf{c}$ can be found in Appendix G.3. Given the above random permutation model σ and the user t with corresponding $\mathbf{r_t}$ and $\mathbf{M}_t$, we solve the online allocation problem at each time t and compute the cumulative revenue. Note that we no longer allocate any impressions to advertisements whose resources are depleted, though the corresponding dual variable is continually updated.

6.2. Efficiency of OG-MA

Utilizing the synthetic dataset, we compare OG-MA to the extension of DSDA (Balseiro et al., 2021) in the multi-slots setting. Concretely, we implement DSDA via using CVX package (Diamond & Boyd, 2016) with MOSEK solver (ApS, 2022), and apply the DSDA to directly solve the problem (2). We simulate the scale of the industrial applications by setting (A_t, N) to the moderate size (Huang et al., 2020) and report elapsed time averaging from 10 trials. As can be seen from Figure 2, the OG-MA is $3 \sim 4$ order faster than DSDA with CVX in general. In particular, for the case $(A_t = 100, N = 2)$, the baseline method requires hundreds of seconds to obtain the solution (Note that $T = 10000$ in our experiment), which is already unacceptable in real-world applications. Further, the baseline method takes 3200 seconds to optimize the problem with $(A_t = 1000, N = 10)$. The above results are far beyond satisfactory since an online server has to deal with thousands of requests within seconds. On the opposite, the computation cost is only 1 second for OG-MA given $(A_t = 1000, N = 10)$.

6.3. Inference Efficiency for Different N

Figure 3 plots the elapsed time for real-time inference with different slots N in both the industrial dataset and the synthetic dataset. We choose the regularization level $\alpha = 0.01$ and decay factor $\gamma = \frac{1}{2}$. The number of advertisements A_t for the industrial and the synthetic dataset is set to 20 and 2^{15}, respectively. In the industrial dataset, the left part of Figure 3 shows that the inference procedure is within a few milliseconds where the number of N is relatively small in real-world applications. Besides, in the synthetic dataset, we set N from 1 to 2^{12} to evaluate the complexity for completeness. The right part of Figure 3 clearly shows that our OG-MA algorithm efficiently makes an online allocation, where the complexity grows sub-linearly w.r.t. the number of slots N. In addition, we argue that the input data leads to the gap between empirical (sub-linear) results as shown in Figure 3 and theoretical (linear) results in Section 4.1. Precisely, $O(N + NA_t + A_t \log A_t)$ is the worst complex-

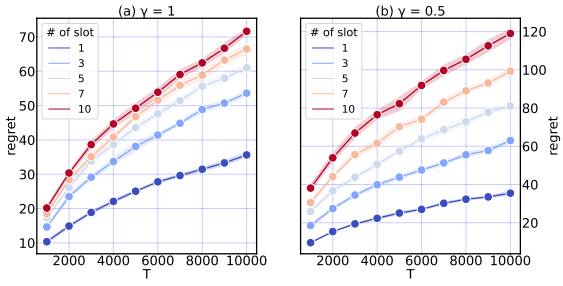

Figure 4: Empirical regret along horizon T for different N.

ity order of our OG-MA algorithm given fully adversarially designed input data.

6.4. Verification of Regret Bound

In this experiment, the regret analysis is validated using the AliExpress dataset. We plot the mean regret with 95% confidence interval over 100 random trials for each value of T and set $\alpha = 0.01$. Along the horizon, our algorithm attains sub-linear regret consistently. As stated in Remark 5.4, the choice of click models affect the total impression capacity, which further determines the order of the upper regret bound. Figure 4 demonstrates regret variations of different hyperparameters γ for the choice of click models. Tuning the number of slots, $\gamma = 1$ leads to $\mathcal{O}(logN)$ regret while $\gamma = \frac{1}{2}$ leads to $\mathcal{O}(\sqrt{N})$ regret, which coincides with the theoretical analysis in Theorem 5.3.

6.5. Trade-off between Revenue and Diversity

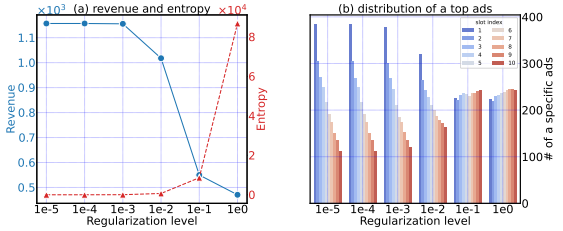

Figure 5: (a) shows the trade-off between entropy and revenue; (b) illustrates the diversity of a top ads allocation.

Figure 5 demonstrates how the revenue and entropy change in different regularization levels and presents the distribution of specific top advertisements allocated in different slots. We choose the regularization level $\alpha \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ in the experiment. The results suggest that higher entropy leads to better diversity in matching while potentially decreasing the revenue. In detail, by setting $\alpha = 0.01$, the algorithm achieves a good trade-off that matching diversity is noticeably improved with acceptable revenue reduction.

7. Conclusion

This paper proposes a new online matching framework for a multi-slots online matching problem. We introduce the high entropy in the matching objective, which generates diversified results, and prove that our algorithm enjoys a sub-linear expected regret bound without constraint violations. Besides, we conduct extensive experiments to validate the superior performance of the proposed framework.

References

Agrawal, S. and Devanur, N. R. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1405–1424. SIAM, 2014.

Agrawal, S., Wang, Z., and Ye, Y. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

Agrawal, S., Zadimoghaddam, M., and Mirrokni, V. Proportional allocation: Simple, distributed, and diverse matching with high entropy. In *International Conference on Machine Learning*, pp. 99–108. PMLR, 2018.

Ahmed, F., Dickerson, J. P., and Fuge, M. Diverse weighted bipartite b-matching. *arXiv preprint arXiv:1702.07134*, 2017.

ApS, M. *MOSEK Fusion API for Python. Release 9.3.20*, 2022. URL https://docs.mosek.com/latest/pythonfusion/index.html.

Arlotto, A. and Gurvich, I. Uniformly bounded regret in the multisecretary problem. *Stochastic Systems*, 9(3): 231–260, 2019.

Balseiro, S., Lu, H., and Mirrokni, V. Dual mirror descent for online allocation problems. In *International Conference on Machine Learning*, pp. 613–628. PMLR, 2020.

Balseiro, S., Lu, H., and Mirrokni, V. Regularized online allocation problems: Fairness and beyond. In *International Conference on Machine Learning*, pp. 630–639. PMLR, 2021.

Chapelle, O. and Zhang, Y. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pp. 1–10, 2009.

Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., and He, X. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240*, 2020.

Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pp. 87–94, 2008.

De Leeuw, J., Hornik, K., and Mair, P. Isotone optimization in r: pool-adjacent-violators algorithm (pava) and active set methods. *Journal of statistical software*, 32:1–24, 2010.

Devanur, N. R. and Hayes, T. P. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 71–78, 2009.

Devanur, N. R., Jain, K., Sivan, B., and Wilkens, C. A. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, pp. 29–38, 2011.

Di Noia, T., Rosati, J., Tomeo, P., and Di Sciascio, E. Adaptive multi-attribute diversity for recommender systems. *Information Sciences*, 382:234–253, 2017.

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2016. URL https://stanford.edu/~boyd/papers/pdf/cvxpy_paper.pdf. To appear.

Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., and Pál, M. Online ad assignment with free disposal. In *International workshop on internet and network economics*, pp. 374–385. Springer, 2009.

Feldman, J., Henzinger, M., Korula, N., Mirrokni, V. S., and Stein, C. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*, pp. 182–194. Springer, 2010.

Guo, F., Liu, C., and Wang, Y. M. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*, pp. 124–131, 2009.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Huang, J.-T., Sharma, A., Sun, S., Xia, L., Zhang, D., Pronin, P., Padmanabhan, J., Ottaviano, G., and Yang, L. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2553–2561, 2020.

Kesselheim, T., Tönnis, A., Radke, K., and Vöcking, B. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 303–312, 2014.

Kveton, B., Szepesvari, C., Wen, Z., and Ashkan, A. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pp. 767–776. PMLR, 2015.

Li, X. and Ye, Y. Online linear programming: Dual convergence, new algorithms, and regret bounds. *arXiv preprint arXiv:1909.05499*, 2019.

Li, X., Sun, C., and Ye, Y. Simple and fast algorithm for binary integer and online linear programming. *arXiv preprint arXiv:2003.02513*, 2020.

Mehta, A., Saberi, A., Vazirani, U., and Vazirani, V. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.

Qin, L. and Zhu, X. Promoting diversity in recommendation by entropy regularizer. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

Talluri, K. and Van Ryzin, G. An analysis of bid-price controls for network revenue management. *Management science*, 44(11-part-1):1577–1593, 1998.

Van Der Vaart, A. W., van der Vaart, A. W., van der Vaart, A., and Wellner, J. *Weak convergence and empirical processes: with applications to statistics*. Springer Science & Business Media, 1996.

Yan, J., Xu, Z., Tiwana, B., and Chatterjee, S. Ads allocation in feed via constrained optimization. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3386–3394, 2020.

Zhang, M. Enhancing diversity in top-n recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pp. 397–400, 2009.

Zhong, W., Jin, R., Yang, C., Yan, X., Zhang, Q., and Li, Q. Stock constrained recommendation in tmall. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2287–2296, 2015.

A. Proof of Proposition 3.1.

The proof of the proposition consists of two components. First, we obtain that the optimal solution $\mathbf{X}_t^*$ to problem (2) is also a feasible solution to the two-step problems (6)(7) because $\mathbf{X}_t^*$ satisfies the domain constraints $\mathcal{Y}$ in problem (6). Then, the inequality holds:

$$\sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{y}_t^* + \alpha \mathcal{H}(\mathbf{y}_t^*) \geq \sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{X}_t^* \mathbf{c} + \alpha \mathcal{H}(\mathbf{X}_t^* \mathbf{c})$$

where $\mathbf{y}_t^*$ is the optimal solution to problem(6). In the next step, we show that the linear equations(7) always have a feasible solution $\mathbf{X}_t$ with given optimal $\mathbf{y}_t^*$. This can be easily shown in the feasibility analysis of the swapping probabilities in Algorithm 3. Consequently, it brings the following inequality:

$$\sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{y}_t^* + \alpha \mathcal{H}(\mathbf{y}_t^*) \leq \sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{X}_t^* \mathbf{c} + \alpha \mathcal{H}(\mathbf{X}_t^* \mathbf{c})$$

Combining above two inequalities, we conclude the equivalent result.

B. Proof of Proposition 3.2

We further introduce $\boldsymbol{\tau}_t$ as the dual variables of domain constraints $\mathcal{Y}$ to obtain the Lagrangian dual formulation:

$$\min_{\boldsymbol{\lambda}\geq 0, \boldsymbol{\tau}\geq 0} \max_{\mathbf{y}\in\mathcal{Y}} \sum_{t=1}^{T} [\mathbf{r}_t^\top \mathbf{y}_t + \alpha \mathcal{H}(\mathbf{y}_t) + \sum_{n=1}^{N} \tau_{t,n} (\sum_{s=1}^{n} c_s - \sup_{\mathbf{E}_t(n)} \{ \sum_{a\in\mathbf{E}_t(n)} y_{t,a} \})] + \boldsymbol{\lambda}^\top (T\mathbf{B} - \sum_{t=1}^{T} \mathbf{M}_t^\top \mathbf{y}_t) \tag{16}$$

With the K.K.T. conditions, we can further obtain that the optimal $\mathbf{y}_t^*$ can be formulated given $\boldsymbol{\lambda}$ and $\boldsymbol{\tau}_t$:

$$y_{t,a}^* = (\sum_{n=1}^{N} c_n) \frac{v_{t,a} \prod_{n=1}^{N-1} P_{t,a,n}}{\sum_a [v_{t,a} * \prod_{n=1}^{N-1} P_{t,a,n}]} \tag{17}$$

where $P_{t,a,n} = \exp(-\frac{\tau_{t,n} \mathbb{I}(a\in\mathbb{S}(\mathbf{E}_t,n))}{\alpha})$ is a penalty term to enforce the cumulative top-n value of $\mathbf{y}_t^*$ under domain constraints $\mathcal{Y}$ and $\mathbb{I}(\cdot)$ is the indicator function. Here, $\mathbb{S}(\mathbf{y}_t, n)$ represents the top-n elements in $\mathbf{y}_t$.

We first prove that the top-n elements inverted by (9) are the top-n elements of optimal $\mathbf{y}_t^*$ in the first half of Proposition 3.2. By contradiction, if there exists $v_{t,i} > v_{t,j}$ and $y_{t,i}^* < y_{t,j}^*$, then we have $P_{t,i,n} > P_{t,j,n}, \forall n < N$ by the definition of $P_{t,a,n}$ and non-negative properties of $\boldsymbol{\tau}_t$. This incurs $v_{t,i} \prod_{n=1}^{N-1} P_{t,i,n} > v_{t,j} \prod_{n=1}^{N-1} P_{t,j,n}$ given $v_{t,i} > v_{t,j}$. Bring it into the formula (17), we have $y_{t,i}^* > y_{t,j}^*$ which is conflict with the assumption. Hence, we have $y_{t,a}^*$ in the same order as $v_{t,a}$.

In the next step, we show that the $e_{t,i}^*$ also in the same order as $v_{t,a}$. Using the above result, $P_{t,a,n}$ in (17) can be further represented as $P_{t,a,n} = \exp(-\frac{\tau_{t,n} \mathbb{I}(a\leq n)}{\alpha})$. For any $y_{t,i}^* \geq y_{t,j}^*$, we have $P_{t,i,n} \leq P_{t,j,n} \forall n$, implying $\prod_{n=1}^{N-1} P_{t,i,n} \leq \prod_{n=1}^{N-1} P_{t,j,n}$. Furthermore, the following holds:

$$\frac{\sum_a [v_{t,a} * \prod_{n=1}^{N-1} P_{t,a,n}]}{\prod_{n=1}^{N-1} P_{t,i,n}} \geq \frac{\sum_a [v_{t,a} * \prod_{n=1}^{N-1} P_{t,a,n}]}{\prod_{n=1}^{N-1} P_{t,j,n}}$$

which implies $e_{t,i}^* \geq e_{t,j}^*$ and the proof is concluded.

C. Proof of Proposition 3.4.

Using the result in Proposition 3.2, we conclude that the efficiency value $e_{t,a}$ and contribution value $v_{t,a}$ are in the same order for an optimal solution $\mathbf{y}_t^*$. Then, we gather elements together if $e_{t,a} = e_{t,a+1}$ for all $a \in \mathbf{E}_a$ sorted by $v_{t,a}$, and thus we can formulate a unique optimal block set $\{\mathbf{B}_r\}^*$ as defined in Definition 3.3. In each block, all elements share the same efficiency and the poof is concluded.

D. Proof of Proposition 3.5.

As stated in Proposition 3.5, if $y_{t,r}, y_{t,r+1}$ belong to different blocks in the optimal block set $\{\mathbb{B}_r\}^*$, then we have:

$$\begin{aligned}
E(\mathbb{B}^*_{(p,r]}) &= \frac{\sum_{p<a\leq r} v_{t,a}}{\sum_{p<a\leq r} y^*_{t,a}} \\
&= \frac{\sum_a [v_{t,a} * \prod_{n=1}^{N-1} P_{t,a,n}]}{\sum_{n=1}^{N} c_n} * \frac{\sum_{p<a\leq r} v_{t,a}}{\sum_{p<a\leq r}[v_{t,a} * \prod_{n=1}^{N-1} P_{t,a,n}]} \\
&\geq \frac{\sum_a [v_{t,a} * \prod_{n=1}^{N-1} P_{t,a,n}]}{\sum_{n=1}^{N} c_n} * \frac{1}{\prod_{n=1}^{N-1} P_{t,r,n}} \\
&> \frac{\sum_a [v_{t,a} * \prod_{n=1}^{N-1} P_{t,a,n}]}{\sum_{n=1}^{N} c_n} * \frac{1}{\prod_{n=1}^{N-1} P_{t,r+1,n}} \\
&\geq \frac{\sum_{r<a\leq q} v_{t,a}}{\sum_{r<a\leq q} y^*_{t,a}} \\
&= E(\mathbb{B}^*_{(r,q]})
\end{aligned} \tag{18}$$

The first and last equations come from the definition of block's efficiency in Definition (10). By substituting (17) into the first equation, we derive the second equation. The third, fourth, and fifth lines hold based on the analysis of the penalty term $P_{t,a,n}$ in Proposition 3.2.

E. Proof of Theorem 5.1.

Following Proposition 3.2, we consider the minimum dual problem (8) with fixed $\boldsymbol{\lambda}$. In particular, we have the following results by complementary slackness:

$$\begin{cases} \tau^*_{t,n} = 0, & \sum_{s=1}^{n} c_s < \sup\limits_{\mathbf{E}_t(n)} \{\sum_{a\in\mathbf{E}_t(n)} y^*_{t,a}\} \\ \tau^*_{t,n} > 0, & \sum_{s=1}^{n} c_s = \sup\limits_{\mathbf{E}_t(n)} \{\sum_{a\in\mathbf{E}_t(n)} y^*_{t,a}\}. \end{cases} \tag{19}$$

Let $\Delta^*_t = \{a_1, a_2, ..., a_{L_t-1} | a_1 < a_2 ... < a_{L_t-1}\}$ be the index set with $L_t - 1$ indices which satisfies $\tau^*_{t,a} > 0, a \in \Delta^*_t$. After the elements $a \in \mathbf{E}_t$ is inverted by $v_{t,a}$, it's easy to find that Δ^*_t constitutes $L_t - 1$ cut points of $\mathbf{E}_t$ that can constitute a disjoint block set $\{\mathbb{B}_r\}^*$ with L_t blocks. Each block $\mathbb{B}^*_r$ can be represented as $\mathbb{B}^*_{(a_{r-1},a_r]}$. Then, the following efficiency equation holds:

$$E(\mathbb{B}^*_{(a_{r-1},a_r]}) = \frac{\sum_{a_{r-1}<a<a_r} v_{t,a}}{\sum_{a_{r-1}<a\leq a_r} y^*_{t,a}} = \frac{\sum_{a_{r-1}<a<a_r} v_{t,a}}{\sum_{a_{r-1}<a\leq a_r} \mathbb{I}(1 \leq a \leq N) c_a} \tag{20}$$

We prove the block set solved by Algorithm 2 contain the same cut points with Δ^*_t by contradiction. If the block set $\{\mathbb{B}_r\}$ solved by Algorithm 2 is not the optimal block, then one of the following two conditions must be met:

- There exist a block $\mathbb{B}_{(p,q]} \in \{\mathbb{B}_r\}$, and an inner point $r \in (p, q]$ belongs to the optimal separator points Δ^*_t.
- There exist two Blocks $\mathbb{B}_{(p,r]}, \mathbb{B}_{(r,q]} \in \{\mathbb{B}_r\}, p < r < q$ that need to be merged as one of the optimal blocks $\{\mathbb{B}_r\}^*$.

If the first condition holds, then there must be a merge step in Algorithm 2 that leads to the follows:

$$\frac{\sum_{p<a\leq r} v_{t,a}}{\sum_{p<a\leq r} \mathbb{I}(1 \leq a \leq N) c_a} = E(\mathbb{B}_{(p,r]}) \leq E(\mathbb{B}_{(r,q]}) = \frac{\sum_{r<a\leq q} v_{t,a}}{\sum_{r<a\leq q} \mathbb{I}(1 \leq a \leq N) c_a}. \tag{21}$$

Besides, r belongs to the optimal separator points Δ^*_t, it holds that

$$\sum_{0<a\leq r} y^*_{t,a} = \sum_{0<a\leq r} \mathbb{I}(1 \leq a \leq N) c_a \tag{22}$$

and $\forall 0 \leq p < r < q \leq N$:

$$\sum_{p<a\leq r} y^*_{t,a} \geq \sum_{p<a\leq r} \mathbb{I}(1 \leq a \leq N) c_a \tag{23}$$

$$\sum_{r<a\leq q} y^*_{t,a} \leq \sum_{r<a\leq q} \mathbb{I}(1\leq a\leq N)c_a \tag{24}$$

By substituting (23) and (24) into (21), we can obtain $E(\mathbb{B}^*_{(p,r]}) \leq E(\mathbb{B}^*_{(r,q]})$ which is conflicted with Proposition 3.5.

Similarly, if the second condition holds, we have the following inequality held by the stop condition in Algorithm 2:

$$\frac{\sum_{p<a\leq r} v_{t,a}}{\sum_{p<a\leq r}\mathbb{I}(1\leq a\leq N)c_a} = E(\mathbb{B}_{(p,r]}) > E(\mathbb{B}_{(r,q]}) = \frac{\sum_{r<a\leq q} v_{t,a}}{\sum_{r<a\leq q}\mathbb{I}(1\leq a\leq N)c_a}. \tag{25}$$

Besides, if $\mathbb{B}^*_{(p,q]}$ is an optimal block, then it holds that

$$\sum_{p<a\leq r} y^*_{t,a} \leq \sum_{p<a\leq r} c_a\mathbb{I}(1\leq a\leq N) \tag{26}$$

Then we can obtain the following inequalities by combining (25) and (26) :

$$E(\mathbb{B}^*_{(p,r]}) = \frac{\sum_{p<a\leq r} v_{t,a}}{\sum_{p<a\leq r} y^*_{t,a}} \geq \frac{\sum_{p<a\leq r} v_{t,a}}{\sum_{p<a\leq r}\mathbb{I}(1\leq a\leq N)c_a} > \frac{\sum_{p<a\leq q} v_{t,a}}{\sum_{p<a\leq q}\mathbb{I}(1\leq a\leq N)c_a} = E(\mathbb{B}^*_{(p,q]}). \tag{27}$$

It is conflicted with Proposition 3.4. Hence, we have the block set solved by Algorithm 2 containing the same cut points as Δ^*_t. Moreover, the efficiency value $E(\mathbb{B}_r)$ is equal to $E(\mathbb{B}^*_r)$ for all r r.t. by (21). This proves that $\{\mathbb{B}_r\}$ is optimal, and further obtains the formulation of the convex optimization of $\boldsymbol{\lambda}$ in Theorem 5.1.

F. Proof of Theorem 5.3

For a better description, we define $f_t(\mathbf{y}_t) := \mathbf{r}_t^\top \mathbf{y}_t + \alpha\mathcal{H}(\mathbf{y}_t)$ and $|c| := \sum_{n=1}^N c_n$, and it holds that:

$$|f|^{\max} = \max_{\mathbf{y}\in\mathcal{Y}} f_t(\mathbf{y}) \leq |c|(\bar{r} + \alpha\log A), \forall t\in\mathbb{T};\ |f|^{\min} = \min_{\mathbf{y}\in\mathcal{Y}} f_t(\mathbf{y}) \geq 0, \forall t\in\mathbb{T} \tag{28}$$

where A is the number of advertisements. We show that our regret analysis in this part is valid for general concave objective functions $f_t(\mathbf{y}_t)$ with finite upper and lower bounds.

First, we give the following lemma that states the dual space of the optimal $\boldsymbol{\lambda}$ under random order σ.

Lemma F.1. *For any s requests selected from the sample set $\{(\mathbf{r}_t, \mathbf{M}_t)\}_{t=1}^T$, $\boldsymbol{\lambda}^*_s$ is the optimal dual solution of the dual problem:*

$$\min_{\boldsymbol{\lambda}_s\geq 0}\max_{\mathbf{y}\in\mathcal{Y}} \sum_{t=1}^s f_t(\mathbf{y}_t) + \boldsymbol{\lambda}_s^\top(s\mathbf{B} - \sum_{t=1}^s \mathbf{M}_t^\top\mathbf{y}_t)$$

, then it holds that:

$$||\boldsymbol{\lambda}^*_s||_1 \leq \frac{|f|^{\max}}{\underline{b}}.$$

Proof. For any s requests $\{(\mathbf{r}_t, \mathbf{M}_t)\}_{t=1}^s$, if $\boldsymbol{\lambda}^*_s$ denotes the optimal dual solution, we have:

$$\begin{aligned}\sum_{t=1}^s f_t(\mathbf{y}_t(\boldsymbol{\lambda}^*_s)) &= \sum_{t=1}^s f_t(\mathbf{y}_t(\boldsymbol{\lambda}^*_s)) + (\boldsymbol{\lambda}^*_s)^\top\sum_{t=1}^s(\mathbf{B} - \mathbf{M}_t^\top\mathbf{y}_t(\boldsymbol{\lambda}^*_s)) \\ &\geq \sum_{t=1}^s f_t(\hat{\mathbf{y}}_0) + (\boldsymbol{\lambda}^*_s)^\top\sum_{t=1}^s(\mathbf{B} - \mathbf{M}_t^\top\hat{\mathbf{y}}_0) \\ &\geq s(|f|^{\min}) + s\underline{b}||\boldsymbol{\lambda}^*_s||_1.\end{aligned} \tag{29}$$

where $\hat{\mathbf{y}}_0$ denotes a decision matrix with entries equal to 0 for all advertisements (except the advertisements with zero revenue and resource consumption), and thus we have $\mathbf{M}_t^\top\hat{\mathbf{y}}_0 = 0, \forall t\in\mathbb{T}$. The first inequality comes from the maximum properties of $\mathbf{y}_t(\boldsymbol{\lambda}^*_s)$ in the dual problem (8). The second inequality comes from the definition of $|f|^{\min}$ and $\underline{b}$. Applying $|f|^{\max}$ to the left half of inequalities, we can obtain the result in this lemma.

In the next part, we decompose the proof of Theorem 5.3 into three steps. First, although our algorithm can not exceed the resource constraints when making decisions, we prove that the loss of primal objective is well bound after the resources

are depleted. Second, we consider the cumulative revenue computed by solution $\hat{\mathbf{y}}_t$ in Algorithm 2 without the concern of resource violation. The revenue $R(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_T|\sigma)$ is independent of the random variable θ in Algorithm 3. Let $\tau_{\mathcal{A}}$ denote the stopping time when the resources are depleted, and we can further bound the regret by:

$$\text{Regret}(\mathcal{A}|\sigma) \leq (R^* - R(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_T)) + (|f|^{\max} - |f|^{\min})\mathbb{E}_\theta[T - \tau_{\mathcal{A}}] \tag{30}$$

Then, we present the bounds of the above two parts respectively and finally put them together.

Part1. The expected loss by resource violation.

Through the following lemma, we give a bound of $\mathbb{E}_\theta[T - \tau_{\mathcal{A}}]$:

Lemma F.2. *Consider the projected subgradient descent in Algorithm 1 with gradient* $\mathbf{g}(\boldsymbol{\lambda}_t) = \mathbf{B} - \mathbf{M}_t^\top \hat{\mathbf{y}}_t$ *and update rule* $\boldsymbol{\lambda}_{t+1} = Proj_{\boldsymbol{\lambda}\geq 0}\{\boldsymbol{\lambda}_t - \eta \mathbf{g}(\boldsymbol{\lambda}_t)\}$. *Suppose* $\boldsymbol{\lambda}_0 = \mathbf{0}$, *then it holds that*

$$||\boldsymbol{\lambda}_t||_\infty \leq \frac{|f|^{\max}}{\underline{b}} + \eta\overline{m}, \forall t \leq T.$$

Furthermore, the following inequality can bound the expectation of $T - \tau_{\mathcal{A}}$ *given random variable* θ *by:*

$$\mathbb{E}_\theta[T - \tau_{\mathcal{A}}] \leq 2\frac{\overline{m}}{\underline{b}} + \frac{|f|^{\max}}{\underline{b}^2\eta}. \tag{31}$$

Proof. The lemma can be proved by citing Proposition 2 in (Balseiro et al., 2020) under the gradient descent framework. Besides, the random variable θ in the real-time decision is independent of the request input model, so we can obtain that the analysis of the expected stopping time $\mathbb{E}_\theta[T - \tau_{\mathcal{A}}]$ in (Balseiro et al., 2020) still holds under the random permutation model.

Part2. The primal performance.

In the stochastic input model, the coming unobserved requests achieve the same expected primal objective with the total T requests under the t-th updated dual variables $\boldsymbol{\lambda}_t$. In the random permutation model, there exists a gap between the above two expectations. In this part, we give a bound on the gap, then we prove that the updated dual solution $\boldsymbol{\lambda}_t$ in each iteration t is too bad dual solution for the coming $T - t$ requests. Specifically, we present the following two lemmas:

Lemma F.3. *For any* s *requests arrive in a random order from total requests* $\{(\mathbf{r}_t, \mathbf{M}_t)\}_{t=1}^T$, *we denote their optimal primal objective as*

$$R_s^* = \max_{\mathbf{y}\in\mathcal{Y}} \sum_{t=1}^{s} f_t(\mathbf{y}_t).$$

Then, it holds that:

$$\mathbb{E}_\sigma[\frac{1}{T}R_T^* - \frac{1}{s}R_s^*] \leq \frac{|f|^{\max}}{\underline{b}} \frac{\sqrt{2\overline{m}}\log s}{\sqrt{s}} + \frac{|f|^{\max}K}{s} \tag{32}$$

Proof. We denote the dual formulation of arriving s requests as $L_s(\boldsymbol{\lambda}) = \sum_{t=1}^s [f_t(\mathbf{y}_t(\boldsymbol{\lambda})) + \boldsymbol{\lambda}^\top(B - \mathbf{M}_t\mathbf{y}_t)]$. Then, we assume $\boldsymbol{\lambda}_s^*$ as the optimal dual solution of R_s^*, and it holds that:

$$\begin{aligned}
\frac{1}{s}R_s^* = \frac{1}{s}L_s(\boldsymbol{\lambda}_s^*) &= \frac{1}{s}\max_{\mathbf{y}\in\mathcal{Y}}\min_{\boldsymbol{\lambda}\geq 0}\sum_{t=1}^{s}[f_t(\mathbf{y}_t) + (\boldsymbol{\lambda})^\top(B - \mathbf{M}_t\mathbf{y}_t)] \\
&= \frac{1}{s}\max_{\mathbf{y}\in\mathcal{Y}}\sum_{t=1}^{s}[f_t(\mathbf{y}_t) + (\boldsymbol{\lambda}_s^*)^\top(B - \mathbf{M}_t\mathbf{y}_t)] \\
&\geq \frac{1}{s}\sum_{t=1}^{s}[f_t(\mathbf{y}_t(\boldsymbol{\lambda}_T^*)) + (\boldsymbol{\lambda}_s^*)^\top(B - \mathbf{M}_t\mathbf{y}_t(\boldsymbol{\lambda}_T^*)] \\
&\geq \frac{1}{s}\sum_{t=1}^{s} f_t(\mathbf{y}_t(\boldsymbol{\lambda}_T^*)) + (\boldsymbol{\lambda}_s^*)^\top(\mathbf{B} - \frac{1}{s}\mathbf{M}_t\mathbf{y}_t(\boldsymbol{\lambda}_T^*))
\end{aligned} \tag{33}$$

In the next step, we prove that for any s request selected from the total samples $\{(\mathbf{r}_t, \mathbf{M}_t)\}_{t=1}^T$, the resource violation can be well controlled with high probability while the primal solution is recovered by $\boldsymbol{\lambda}_T^*$. Concretely, for each resource constraint

k, the following inequalities hold:

$$\begin{aligned}
&\mathbb{P}(\frac{1}{s}\sum_{t=1}^{s}\mathbf{m}_{t,k}^{\top}\mathbf{y}_t(\boldsymbol{\lambda}_T^*) - B_k \geq d) \\
&\leq \mathbb{P}(\frac{1}{s}\sum_{t=1}^{s}\mathbf{m}_{t,k}^{\top}\mathbf{y}_t(\boldsymbol{\lambda}_T^*) - \frac{1}{T}\sum_{t=1}^{T}\mathbf{m}_{t,k}^{\top}\mathbf{y}_t(\boldsymbol{\lambda}_T^*) \geq d) \\
&\leq \exp\left(-\frac{s^2d^2}{s(\overline{m})^2/2 + d\overline{m}}\right)
\end{aligned} \tag{34}$$

where the first inequality is because the optimal dual solution $\boldsymbol{\lambda}_T^*$ is feasible with $\frac{1}{T}\sum_{t=1}^{T}\mathbf{m}_{t,k}^{\top}\mathbf{y}_t(\boldsymbol{\lambda}_T^*) \leq B_k$, and the second inequality comes from Hoeffding-Bernstein's Inequality for sampling without replacement (Van Der Vaart et al., 1996). By choosing the gradient violation threshold $d = \sqrt{2}\overline{m}\frac{log(s)}{\sqrt{s}}$, it holds that:

$$\exp\left(-\frac{s^2d^2}{s(\overline{m})^2/2 + d\overline{m}}\right) \leq \frac{1}{s}.$$

Taking the union of the above bounds with all K constraints, we readily obtain:

$$\mathbb{P}(\{\exists 1 \leq k \leq K, \frac{1}{s}\sum_{t=1}^{s}\mathbf{m}_{t,k}^{\top}\mathbf{y}_t(\boldsymbol{\lambda}_T^*) - B_k \geq \sqrt{2}\overline{m}\frac{log(s)}{\sqrt{s}}\}) \leq \frac{K}{s} \tag{35}$$

Finally, we obtain the bound in this lemma:

$$\begin{aligned}
\mathbb{E}_\sigma[\frac{1}{T}R_T^* - \frac{1}{s}R_s^*] &\leq \mathbb{E}_\sigma[\frac{1}{T}\sum_{t=1}^{T}f_t(\mathbf{y}_t(\boldsymbol{\lambda}_T^*)) - \frac{1}{s}\sum_{t=1}^{s}f_t(\mathbf{y}_t(\boldsymbol{\lambda}_T^*)) + (\boldsymbol{\lambda}_s^*)^{\top}(\frac{1}{s}\mathbf{M}_t\mathbf{y}_t(\boldsymbol{\lambda}_T^*) - \mathbf{B})] \\
&= \mathbb{E}_\sigma[(\boldsymbol{\lambda}_s^*)^{\top}(\frac{1}{s}\mathbf{M}_t\mathbf{y}_t(\boldsymbol{\lambda}_T^*) - \mathbf{B})]] \\
&\leq \mathbb{E}_\sigma[||\boldsymbol{\lambda}_s^*||_1]\frac{\sqrt{2}\overline{m}\log(s)}{\sqrt{s}} + \frac{|f|^{\max}K}{s}
\end{aligned} \tag{36}$$

The first inequality comes from (33), and the second line comes from the probability properties in the random permutation model. Applying (35) as a conditional probability to the expectation, we can obtain the third inequality. With the bound of $||\boldsymbol{\lambda}_s^*||_1$ and $|f|^{\max}$, we finish the proof.

Lemma F.4. *Consider the projected subgradient descent step in Algorithm 1. Suppose $\boldsymbol{\lambda}_0 = \mathbf{0}$, then it holds that:*

$$\sum_{t=1}^{T}\boldsymbol{\lambda}_t^{\top}\mathbf{g}(\boldsymbol{\lambda}_t) \leq \frac{K(\overline{m}^2 + \overline{b}^2)}{2}\eta T \tag{37}$$

Proof. For any input random order σ, it holds that:

$$\begin{aligned}
&\sum_{t=1}^{T}\boldsymbol{\lambda}_t^{\top}\mathbf{g}(\boldsymbol{\lambda}_t) = \sum_{t=1}^{T}[(\boldsymbol{\lambda}_t^{\top} - \boldsymbol{\lambda}_{t+1}^{\top})\mathbf{g}(\boldsymbol{\lambda}_t) + \boldsymbol{\lambda}_{t+1}^{\top}\mathbf{g}(\boldsymbol{\lambda}_t)] \\
&= \sum_{t=1}^{T}[\eta||\mathbf{g}(\boldsymbol{\lambda}_t)||_2^2 + \boldsymbol{\lambda}_{t+1}^{\top}(\frac{\boldsymbol{\lambda}_t^{\top} - \boldsymbol{\lambda}_{t+1}^{\top}}{\eta})] \\
&= \sum_{t=1}^{T}[\eta||\mathbf{g}(\boldsymbol{\lambda}_t)||_2^2 + \frac{1}{2\eta}(||\boldsymbol{\lambda}_t||_2^2 - ||\boldsymbol{\lambda}_{t+1}||_2^2 - ||\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_{t+1}||_2^2)] \\
&= \sum_{t=1}^{T}\frac{\eta}{2}||\mathbf{g}(\boldsymbol{\lambda}_t)||_2^2 + \frac{1}{2\eta}||\boldsymbol{\lambda}_0||_2^2 - ||\boldsymbol{\lambda}_{T+1}||_2^2 \\
&\leq \frac{K(\overline{m}^2 + \overline{b}^2)}{2}\eta T
\end{aligned} \tag{38}$$

The first four equations come from the update rule of gradient descent and combination of cumulative term. The last

inequality is bounded by the $l2$-norm of the gradient in Assumption 5.2.

Using the above two lemmas, we have the bound of primal performance:

$$\begin{aligned} R^* - R(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_T|\sigma) &= \mathbb{E}_\sigma[T\frac{1}{T}R_T^* - \sum_{t=1}^{T} f_t(\mathbf{y}_t)] \\ &= \mathbb{E}_\sigma[\sum_{t=1}^{T}[\frac{1}{T}R_T^* - \frac{1}{t}R_t^*]] + \mathbb{E}_\sigma[\sum_{t=1}^{T}[\frac{1}{t}R_t^* - f_t(\mathbf{y}_t)]] \\ &= \mathbb{E}_\sigma[\sum_{t=1}^{T}[\frac{1}{T}R_T^* - \frac{1}{t}R_t^*]] + \mathbb{E}_\sigma[\sum_{t=1}^{T}[\frac{1}{T-t+1}R_{T-t+1}^*(t, t+1, ..., T) - f_t(\mathbf{y}_t)]] \end{aligned} \tag{39}$$

where $R_{T-t+1}^*(t, t+1, ..., T)$ represents the optimal objective among the coming unobserved $T-t+1$ requests. The first part of the last inequality can be bound by Lemma F.3. For the second part of the last inequality, we have:

$$\begin{aligned} \mathbb{E}_\sigma[\sum_{t=1}^{T}[\frac{1}{\Delta}R_\Delta^*(t, t+1, ..., T) - f(\mathbf{y}_t)]] &\leq \mathbb{E}_\sigma[\sum_{t=1}^{T}[\frac{1}{\Delta}\sum_{s=t}^{T}[f_s(\mathbf{y}_s(\boldsymbol{\lambda}_t)) + \boldsymbol{\lambda}_t^\top(\mathbf{B} - \mathbf{y}_s(\boldsymbol{\lambda}_t))] - f(\mathbf{y}_t(\boldsymbol{\lambda}_t))]] \\ &\leq \mathbb{E}_\sigma[\sum_{t=1}^{T}[\frac{1}{\Delta}\sum_{s=t}^{T}\boldsymbol{\lambda}_t^\top(\mathbf{B} - \mathbf{y}_s(\boldsymbol{\lambda}_t))]] \\ &= \mathbb{E}_\sigma[\sum_{t=1}^{T}\boldsymbol{\lambda}_t\mathbf{g}(\boldsymbol{\lambda}_t)] \end{aligned} \tag{40}$$

where $\Delta := T - t + 1$. The first inequality comes from the minimum properties of the optimal dual objective. The second inequality and the third equation hold because the coming unobserved $T-t+1$ requests are independent of the dual solution $\boldsymbol{\lambda}_t$.

Applying the results in Lemma F.3 and Lemma F.4 to (39) and (40), we can bound the primal performance:

$$R^* - R(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_T|\sigma) \leq \frac{|f|^{\max}}{\underline{b}}2\sqrt{2\overline{m}}\log T\sqrt{T} + |f|^{\max}K(\log T + 1) \tag{41}$$

In the above inequality, we apply the results that $\sum_{t=1}^{T} 1/T \leq \log T + 1$ and $\sum_{t=1}^{T} 1/\sqrt{T} \leq 2\sqrt{T}$. Finally, we finish the proof of Theorem 5.3 by putting the results together from (31), (41), and (30).

G. Additional details on the numerical experiments

G.1. Synthetic Dataset

The data are generated from the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2)$ and $\boldsymbol{\sigma} = 0.5$. For a varying number of slots, we employ the setting: $\boldsymbol{\beta} = [\underbrace{0, \ldots, 0}_{0.2\text{N}}, \underbrace{2, \ldots, 2}_{0.3\text{N}}, \underbrace{0, \ldots, 0}_{0.2\text{N}}, \underbrace{2, \ldots, 2}_{0.3\text{N}}]$, where N is the number of slots. And $\mathbf{X} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2)$ with $\boldsymbol{\sigma} = 0.1$. In the sampling procedure, we keep the sample if its value is in $[0, 1]$, otherwise, we discard it.

G.2. Estimated Revenue

We employed a two-layers Multi-Layer Perceptron (MLP) to parameterize the revenue function $f(t, a)$. Specifically, we set embedding size $d_{emb} = 8$ for each sparse feature, learning rate $= 0.001$, batch size $= 256$, l2 regularizer $= 0.001$, and select Adam as the optimizer. The two-layers MLP is of [32, 16] hidden units, and its parameters are initialized with HeNormal initializer (He et al., 2015) of a standard deviation of 0.02.

G.3. Click Model

For simplicity, we adopt the position-based model (Craswell et al., 2008) that the impression capacity vector $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^N$ follows a decay probability of $c_n = 1/n^\gamma$, indicating that users have a higher probability examining the top slots than the bottom slots. Other click models (Guo et al., 2009; Chapelle & Zhang, 2009; Kveton et al., 2015) can be applied as well in the proposed algorithm.