Online Optimization of Video-Ad Allocation*

Hanna Sumita†, **Yasushi Kawase**‡§, **Sumio Fujita**♮, **Takuro Fukunaga**†
†National Institute of Informatics
‡Tokyo Institute of Technology
§RIKEN AIP Center
♮Yahoo! JAPAN Research
†{sumita, takuro}@nii.ac.jp ‡kawase.y.ab@m.titech.ac.jp ♮sufujita@yahoo-corp.jp

Abstract

In this paper, we study the video advertising in the context of internet advertising. Video advertising is a rapidly growing industry, but its computational aspects have not yet been investigated. A difference between video advertising and traditional display advertising is that the former requires more time to be viewed. In contrast to a traditional display advertisement, a video advertisement has no influence over a user unless the user watches it for a certain amount of time. Previous studies have not considered the length of video advertisements, and time spent by users to watch them. Motivated by this observation, we formulate a new online optimization problem for optimizing the allocation of video advertisements, and we develop a nearly $(1 - 1/e)$-competitive algorithm for finding an envy-free allocation of video advertisements.

1 Introduction

Internet advertising is one of the main marketing tools today. According to one report [Interactive Advertising Bureau, 2015], the annual revenue in 2014 from internet advertisements (hereafter "ads") in the US was $49.5 billion, which was higher than the total revenue from radio, newspapers, and magazines. Internet ads have also attracted attention from the research community. Numerous computation problems arising from internet advertising have been studied (e.g., [Radovanovic and Heavlin, 2012; Bhalgat *et al.*, 2012; Bharadwaj *et al.*, 2012; Bhalgat *et al.*, 2014; Ieong *et al.*, 2014; Bateni *et al.*, 2014; Balseiro *et al.*, 2014; Hojjat *et al.*, 2014]), and the results from these studies constitute a rich area in computer science.

Among the many kinds of internet display ads, this paper focuses on video-ads. Nowadays video advertising is often used in video streaming services, and it is a rapidly growing industry. One study [Pew Research Center, 2014] estimates that video advertising will make up 15% of the total internet advertising market by 2017. To our knowledge, however, computation problems related to video advertising have not yet been investigated.

A difference between video-ads and traditional display ads is that the former need to consider the time to be viewed by a user. Each video-ad has a time length, and video-ads need to be watched for a certain amount of time to influence users. This is in contrast to traditional display ads. Moreover, the time spent watching video-ads depends on the user's particular situation and interest. When long video-ads for sporting goods are allocated to someone who is busy and uninterested in sports, this user is likely to stop watching the video and leave the website. On the other hand, a user who is not busy and likes sports is likely to watch such video-ads for sporting goods. In addition, this user may watch more than one ad. Indeed, some video streaming services show several ads in succession to a user who is going to watch a long video, such as a movie and a TV show. However, these two factors—the length of a video-ad and the time spent watching it by users—have not been considered in previous studies on display advertising. Hence, algorithms to optimize ad-allocations for traditional display ads are inefficient for video-ads. Motivated by this observation, this paper offers an initial study of the optimization of video-ad allocations.

1.1 Our Contributions

The aim of this paper is to design efficient algorithms for deciding video-ad allocation. Our contributions are summarized as follows:

- We formulate the *video-ad allocation problem*, which is an extension of the *ad-auction problem* introduced by Mehta *et al.* [2007].
- We present an online algorithm for the video-ad allocation problem, by showing that the video-ad allocation problem is included by a general online allocation framework proposed by Goel *et al.* [2010]. In addition, we analyze the dependence of its competitive ratio on the ratio of bids to budgets of advertisers, which was ignored in the analysis of Goel *et al.*
- We also consider envy-free pricing and user-dependent video-lengths in the video-ad allocation problem for more practical modeling. To obtain a $(1 - 1/e)$-competitive algorithm for this setting, we extend the framework of Goel *et al.*

*This work was supported by JST ERATO Grant Number JPMJER1201, Japan, and JSPS KAKENHI Grant Number JP16K16005.

Formulation of the Video-Ad Allocation Problem

We formulated the video-ad allocation problem, which extends the so-called ad-auction problem proposed by Mehta *et al.* [2007] for traditional display advertising. The ad-auction problem is an online optimization problem of allocating ads to users arriving at sites one-by-one so that the total revenue from advertisers is maximized. When a user arrives, all advertisers submit a price (or "bid") that they are willing to pay to show his ad to the user. Only one advertiser wins the auction to be assigned to the user. Each advertiser has a budget, and they cannot pay beyond the remaining budget. Mehta *et al.* proposed a $(1 - 1/e)$-competitive online algorithm for the ad-auction problem under the small bid assumption—i.e., each bid is relatively smaller than the associated budgets. Their analysis uses a technique called factor-revealing linear programming (LP). Subsequently, Buchbinder *et al.* [2007] proposed another $(1 - 1/e)$-competitive online algorithm based on the primal-dual method.

For the video-ad allocation problem, we introduce two factors to the ad-auction problem: video length and viewing time. Each advertiser has a video-ad with its length, and each user has a viewing limit (or "capacity") representing the extent of that user's viewing time. We assume that we know the viewing capacity on arrival of a user. While the ad-auction problem allocates only one ad to each user's ad slot, the video-ad allocation problem can show more than one video-ad to each user in succession in the user's ad slot. For simplicity, we first assume that a user watches allocated video-ads to the end, as long as the total length of the ads does not exceed the user's viewing capacity. This constraint has the same structure as the knapsack problem, which is a classical optimization problem. This makes the problem much more difficult than the ad-auction problem.

Relationship with the Framework of Goel et al.

We present an online algorithm for the video-ad allocation problem. The main ingredient in this algorithm is a general framework of the ad-auction problem proposed by Goel *et al.* [2010] in a context of the ad-auction problem with the generalized second-price scheme. In their model, it is allowed to allocate more than one ad to a single user. Instead, the input of the problem specifies sets of ads (called *feasible allocation*) which can be assigned to a user and an associated pricing scheme. Supposing that an algorithm for computing a maximum weight feasible set is available, Goel *et al.* gave an online primal-dual algorithm for this general online problem. Its competitive ratio is $1 - 1/e$ under the small bid assumption. We note that the framework of Goel *et al.* includes the ad-auction problem, and hence the algorithm of Goel *et al.* extends the algorithm of Buchbinder *et al.* [2007].

To apply the algorithm of Goel *et al.* to the video-ad allocation problem, we have to show that a maximum weight feasible set can be computed. Indeed, in the video-ad allocation problem, this computation is equivalent to solving the knapsack problem. This observation indicates that the video-ad allocation problem admits a $(1 - 1/e)$-competitive algorithm under the small bid assumption.

We also analyze the dependence of the competitive ratio of the algorithm on the ratio of bids to budgets of advertisers. Let $R_{\max}$ denote the maximum of ratio of a bid to a budget over all advertisers and their bids to users. The small bid assumption demands that $R_{\max}$ is almost 0. To explain precisely, the competitive ratios of algorithms in [Buchbinder *et al.*, 2007; Mehta *et al.*, 2007; Goel *et al.*, 2010] depend on $R_{\max}$, and approach $1 - 1/e$ when $R_{\max}$ approaches 0. Goel *et al.* did not describe how the competitive ratio of their algorithm depends on $R_{\max}$ explicitly. We present an explicit description of this dependence.

Envy-Free Pricing and User-Dependent Video Lengths

We also consider envy-free pricing in the video-ad allocation problem. In this setting, we are required to decide pricing for each advertiser together with a video-ad allocation. A pair of a pricing and an allocation is called *envy-free* if the payment of each advertiser is at most those of advertisers with longer allocated ads.

In addition, we assume that the length of a video depends on users. The purpose of this assumption is to model the preferences of users on the video topics. For instance, a user who prefers sports over cosmetics watches videos about sports more likely than ones about cosmetics. We incorporate this phenomenon by assuming that the length of a video is shorter if a user prefers it. If the length is shorter, then a user watches the video more likely because of the capacity constraint.

To deal with this extended setting, we generalize the framework of Goel *et al.* While the framework of Goel *et al.* assumes that the payment of each advertiser is decided from the allocated ads, the payments should also be decision variables in the envy-free pricing. Hence, in our new framework, we assume that feasible pairs of allocation and pricing are specified, and design an online primal-dual algorithm under the assumption that a maximum weight feasible pair can be computed. The competitive ratio of this algorithm is $1 - 1/e$ if the small bid assumption holds.

1.2 Organization

The remainder of this paper is organized as follows. Section 2 introduces the video-ad allocation problem, and explains its relationship with Goel *et al.* Section 3 presents and analyzes our algorithm for the setting with envy-free pricing. Section 4 evaluates our algorithms through computational experiments. Finally, Section 5 concludes the paper.

2 Video-ad Allocation Problem

2.1 Setting

Let $N = \{1, \ldots, n\}$ be a set of n advertisers. Each advertiser $i \in N$ has a budget B_i and a video-ad that is t_i in length. Further, let $M = \{1, \ldots, m\}$ be a set of m users. Each user $j \in M$ has a viewing capacity T_j, representing the extent of time where the user will allow the publisher to show video-ads. Users arrive one-by-one; below, we assume that j denotes the jth arriving user. Upon the arrival of a user j, each advertiser i submits a bid b_{ij}, and an online algorithm allocates a set S_j of advertisers to user j. The allocation is required to satisfy the following capacity constraints which demands that the total ad length from advertisers in S_j does not exceed the viewing capacity of user j—i.e., $\sum_{i \in S_j} t_i \le T_j$ for each $j \in M$.

When an advertiser i is allocated to user j, then i pays b_{ij} from the advertiser's budget. For any user j, the publisher allocates only ads of those who have enough budgets to pay the bids. When the allocation to each user j is represented by the advertiser set $S_j \subseteq N$, the payment of an advertiser i is $\sum_{j \in M : i \in S_j} b_{ij}$. The objective of the algorithm is to maximize the total revenue after all users in M arrive—i.e., $\sum_{i \in N} \sum_{j \in M : i \in S_j} b_{ij}$. We assume without loss of generality that each bid b_{ij} is smaller than the budget B_i, and define $R_{\max}$ as $\max_{i \in N, j \in M} b_{ij}/B_i$ (≤ 1).

We take the adversarial input model, where an online algorithm decides the allocation to a user j without information regarding users arriving after j. When an algorithm computes an allocation by using all information about users, it is called *offline*. Let ALG and OPT denote the revenue for an online algorithm and the best offline algorithm, respectively. For $\beta \leq 1$, the online algorithm is referred to as *β-competitive* if ALG $\geq \beta \cdot$ OPT for any instance. Accordingly, β denotes the *competitive ratio* for an online algorithm if β is the maximum number such that the algorithm is β-competitive.

The ad-auction problem studied by [Buchbinder *et al.*, 2007; Mehta *et al.*, 2007] corresponds to the special case with $t_i = 1, i \in N$ and $T_j = 1, j \in M$.

2.2 General Framework of Goel et al.

Goel *et al.* [2010] introduced an abstract online problem that includes the ad-auction problem. In this subsection, we introduce this problem, and show that the video-ad allocation problem is included in it.

First, we give the problem definition. We are given sets of advertisers and users, where we denote the former by N and the later by M. Each user j specifies a subfamily $\mathcal{C}_j$ of 2^N that comprises feasible allocations of ads to user j. We are also given a budget B_i of each advertiser $i \in N$ and a bid b_{ij} for each advertiser $i \in N$ and user $j \in M$, but the length t_i of video-ads and the viewing capacity T_j of users are not given here. The problem seeks to find $S_j \in \mathcal{C}_j$ for each $j \in M$ such that $\sum_{j \in M : i \in S_j} b_{ij} \leq B_i$ for each $i \in N$. The objective is to maximize $\sum_{j \in M} \sum_{i \in S_j} b_{ij}$.

Goel *et al.* presented an online algorithm for this problem, assuming that following two conditions hold for each $j \in M$:

- $\mathcal{C}_j$ is subset-closed—i.e., if $X \subseteq Y \in \mathcal{C}_j$, then $X \in \mathcal{C}_j$;
- given any non-negative weights δ_i of advertisers $i \in N$, there exists an algorithm for finding $S \in \mathcal{C}_j$ that maximizes $\sum_{i \in S} \delta_i$.

The competitive ratio of their algorithm is $1 - 1/e$ under the small bid assumption.

The video-ad allocation problem is included in this problem. This can be seen by setting $\mathcal{C}_j = \{S \mid \sum_{i \in S} t_i \leq T_j\}$. With this definition, $\mathcal{C}_j$ is downward-closed. Moreover, the problem of finding $S \in \mathcal{C}_j$ that maximizes $\sum_{i \in S} \delta_i$ is equivalent to the knapsack problem, and hence it admits a $(1-\epsilon)$-approximation polynomial-time algorithm for any $\epsilon > 0$ and an exact pseudo-polynomial time algorithm, whose running time depends on n and T_j. This fact indicates the following theorem.

Theorem 1. *Under the small bid assumption, the video-ad allocation problem admits a polynomial-time $(1 - 1/e - \epsilon)$-competitive algorithm for any constant $\epsilon > 0$, and a pseudo-polynomial time $(1 - 1/e)$-competitive algorithm.*

Although the competitive ratio given in the above depends on $R_{\max}$, Goel *et al.* did not describe how it depends on $R_{\max}$ explicitly. In the subsequent section, we analyze the competitive ratio of an algorithm that extends the one of Goel *et al.*, describing its dependence on $R_{\max}$.

3 Envy-Free Pricing and User-Dependent Video Lengths

3.1 Setting

In this section, we discuss an envy-free pricing setting. Here, on arrival of a user, we decide both the allocation of video-ads and the pricing, i.e., the charge for each allocated advertiser.

Fix a user $j \in M$. We denote the charge for an advertiser $i \in N$ by p_i. Then, when user j arrives, an online algorithm is required to decide a pair of a pricing p and a video-ad allocation $S \subseteq N$. This pair is feasible in the envy-free setting if (i) $\sum_{i \in S} t_i \leq T_j$, (ii) $p_i \leq b_{ij}$ for each $i \in N$, (iii) $p_i = 0$ for any $i \notin S$, and (iv) $p_i \geq p_{i'}$ for any $i, i' \in S$ with $t_i \geq t_{i'}$ (the envy-freeness).

In addition, we suppose that the length of an video-ad depends on users to model the phenomenon that a user watches a video more likely if it is about his/her favorite topic. Hence, in this section, we let t_{ij} denote the length of an ad i when it is assigned to a user j. We use this user-dependent length in the capacity constraints (i.e., constraint $\sum_{i \in S} t_i \leq T_j$ is replaced by $\sum_{i \in S} t_{ij} \leq T_j$) while the envy-freeness is defined with regard to the original length.

3.2 New Framework

To deal with envy-free pricing, we generalize the framework of Goel *et al.* by introducing the concept of *outcomes*. For a nonnegative vector $p \in \mathbb{R}_+^N$ and a set $S \subseteq N$, we say that (p, S) is an outcome, meaning that p_i is the charge to advertiser $i \in N$ and S is the allocation.

In our framework, a set $\mathcal{C}_j$ of feasible outcomes is specified for each $j \in M$. We suppose that $\mathcal{C}_j$ satisfies the following properties:

- $\mathcal{C}_j$ is subset-closed—i.e., for any $(p, S) \in \mathcal{C}_j$ and $S' \subseteq S$, it holds $(p', S') \in \mathcal{C}_j$, where $p' \in \mathbb{R}_+^N$ is defined by $p'_i = p_i$ for $i \in S'$ and $p'_i = 0$ for $i \in N \setminus S'$;
- we can obtain an α-approximate solution (p, S) of
$$\max_{(p,S) \in \mathcal{C}_j} \sum_{i \in N} \delta_i p_i \tag{1}$$
for $\alpha \leq 1$ and for any $\delta_i \in [0, 1]$, $i \in N$.

In this section, the definition of $R_{\max}$ is modified to $R_{\max} = \max\{p_i/B_i \mid i \in N,\ j \in M,\ (p, S) \in \mathcal{C}_j\}$.

The envy-free pricing setting can be captured by this new framework by setting $\mathcal{C}_j$ as follows:

$$\mathcal{C}_j = \left\{ (p, S) \;\middle|\; \begin{array}{ll} \sum_{i \in S} t_{ij} \leq T_j, & \\ p_i \leq b_{ij} & (i \in S), \\ p_i = 0 & (i \notin S), \\ p_i \geq p_{i'} & (i, i' \in S,\ t_i \geq t_{i'}) \end{array} \right\}. \tag{2}$$

It is not difficult to see that this $\mathcal{C}_j$ is subset-closed. The following lemma presents a pseudo-polynomial time algorithm for finding an outcome that attains the maximum in (1), i.e., the second property required for $\mathcal{C}_j$ is satisfied with $\alpha = 1$.

Lemma 1. *If $\mathcal{C}_j$ is defined by* (2)*, there exists a pseudo-polynomial time algorithm for finding $(p, S) \in \mathcal{C}_j$ that achieves the maximum in* (1) *for any $\delta_i \in [0, 1]$, $i \in N$.*

Proof. Recall that $N = \{1, \dots, n\}$. We assume without loss of generality that $t_1 \leq \cdots \leq t_n$. For each $i, k \in N$ and $t \in \{1, \dots, T_j\}$, define $v(i, k, t)$ as the maximum discounted revenue $\sum_{i \in N} \delta_i p_i$ achieved by $(p, S) \in \mathcal{C}_j$ such that $S \subseteq \{1, \dots, i\}$, all advertisers are charged at most b_{kj} (i.e., $p_{i'} \leq b_{kj}$ for all $i' \in N$), and the total length of allocated video-ad is at most t (i.e., $\sum_{i' \in S} t_{i'j} \leq t$). In other words, $v(i, k, t)$ is

$$\max \left\{ \sum_{i' \in S} \delta_{i'} p_{i'} \;\middle|\; \begin{array}{l} \sum_{i' \in S} t_{i'j} \leq t, \\ p_{i'} \leq \min\{b_{i'j}, b_{kj}\} \ (i' \in S), \\ p_{i'} \geq p_{j'} \ (i', j' \in S,\ t_{i'} \geq t_{j'}), \\ S \subseteq \{1, \dots, i\} \end{array} \right\}. \quad (3)$$

Note that (1) $= \max_{k \in N} v(n, k, T)$. Let $p(i, k, t)$ and $S(i, k, t)$ be the pricing and the allocation achieving $v(i, k, t)$. For convention, let $v(i, k, t) = 0$, $S(i, k, t) = \emptyset$, and $p(i, k, t)_{i'} = 0$, $i' \in N$ if $i = 0$ or if $t \leq 0$.

Let $i \in N$. If $S(i, k, t)$ does not contain ad i, then we have $S(i, k, t) = S(i-1, k, t)$, $p(i, k, t)_i = 0$, $p(i, k, t)_{i'} = p(i-1, k, t)_{i'}$ for $i' \in N \setminus \{i\}$, and $v(i, k, t) = v(i-1, k, t)$. If $S(i, k, t)$ contains ad i and $b_{ij} \geq b_{kj}$, then we have $S(i, k, t) = S(i-1, k, t - t_{ij}) \cup \{i\}$, $p(i, k, t)_i = b_{kj}$, $p(i, k, t)_{i'} = p(i-1, k, t - t_{ij})_{i'}$ for $i' \in N \setminus \{i\}$, and $v(i, k, t) = v(i-1, k, t - t_{ij}) + \delta_i b_{kj}$. If $S(i, k, t)$ contains ad i and $b_{ij} < b_{kj}$, then we have $S(i, k, t) = S(i-1, i, t - t_{ij}) \cup \{i\}$, $p(i, k, t)_i = b_{ij}$, $p(i, k, t)_{i'} = p(i-1, i, t - t_{ij})_{i'}$ for $i' \in N \setminus \{i\}$, and $v(i, k, t) = v(i-1, i, t - t_{ij}) + \delta_i b_{ij}$, since prices of ads $1, \dots, i-1$ are at most the price of ad i. Because of this case analysis, we obtain a recursive formula of $v(i, k, t)$ as follows: if $b_{kj} \leq b_{ij}$, then

$$v(i, k, t) = \max \{v(i-1, k, t), v(i-1, k, t - t_{ij}) + \delta_i b_{kj}\},$$

and otherwise,

$$v(i, k, t) = \max \{v(i-1, k, t), v(i-1, i, t - t_{ij}) + \delta_i b_{ij}\}.$$

A similar formula holds also for $S(i, k, t)$ and $p(i, k, t)$. Therefore, we can calculate $v(i, k, t)$, $S(i, k, t)$, and $p(i, k, t)$ for all i, k, and t in $O(n^2 T_j)$ time. □

3.3 Algorithm

In this subsection, we present an algorithm for the framework defined in the previous section. We notice that $\mathcal{C}_j$ is an arbitrary set of outcomes that satisfy the above two properties in the following discussion.

For an outcome $c \in \mathcal{C}_j$, we let S_c and p_c denote the allocation and pricing in c, respectively. Our algorithm is based on the following LP relaxation:

$$\begin{array}{lll} \max & \sum_{j \in M} \sum_{c \in \mathcal{C}_j} \sum_{i \in N} p_{ci} x_{cj} & \\ \text{s.t.} & \sum_{j \in M} \sum_{c \in \mathcal{C}_j} p_{ci} x_{cj} \leq B_i & \forall i \in N, \\ & \sum_{c \in \mathcal{C}_j} x_{cj} \leq 1 & \forall j \in M, \\ & x_{cj} \geq 0 & \forall j \in M,\ c \in \mathcal{C}_j. \end{array} \quad (4)$$

For each integer feasible solution for (4), there is a corresponding feasible allocation for the video-ad allocation problem, and they achieve the same objective value in their own problems. Hence, the optimal objective value of (4) is an upper bound on the maximum revenue.

The dual of (4) is written as

$$\begin{array}{lll} \min & \sum_{i \in N} B_i y_i + \sum_{j \in M} z_j & \\ \text{s.t.} & \sum_{i \in N} p_{ci} y_i + z_j \geq \sum_{i \in N} p_{ci} & \forall j \in M,\ c \in \mathcal{C}_j, \\ & y_i \geq 0 & \forall i \in N, \\ & z_j \geq 0 & \forall j \in M. \end{array} \quad (5)$$

Owing to the strong duality of LPs, the optimal objective value of (4) is equal to the optimal objective value of (5).

Our algorithm simultaneously constructs both an integer solution x feasible for (4) and a solution (y, z) feasible for (5). To prove that the algorithm is β-competitive, it suffices to show that these solutions satisfy

$$\sum_{j \in M} \sum_{c \in \mathcal{C}_j} \sum_{i \in N} p_{ci} x_{ij} \geq \beta \left(\sum_{i \in N} B_i y_i + \sum_{j \in M} z_j \right). \quad (6)$$

When our algorithm is invoked, y_i is initialized to 0 for all $i \in N$. When a user j arrives, the algorithm computes an α-approximate solution c_j^* for $\max_{c \in \mathcal{C}_j} \sum_{i \in N} p_{ci}(1 - y_i)$. Without loss of generality, we assume that

$$S_{c_j^*} \text{ does not contain the advertisers } i' \text{ with } y_{i'} \geq 1. \quad (7)$$

The outcome c_j' assigned for j is defined as the one obtained from c_j^* by canceling the allocation of advertisers whose remaining budgets exceeds the assigned prices. Then the algorithm sets z_j to the sum of $\sum_{i \in N} p_{c_j^* i}(1 - y_i)/\alpha$. Updating y_i depends on a parameter $\gamma = (1 + R_{\max}/\alpha)^{1/R_{\max}}$ (> 1). Note that γ approaches $e^{1/\alpha}$ as $R_{\max}$ approaches 0. For every advertiser $i \in N$, we update y_i by

$$y_i^{(j)} \leftarrow y_i^{(j-1)} \left(1 + \frac{1}{\alpha} \cdot \frac{p_{c_j^* i}}{B_i} \right) + \frac{1}{\alpha(\gamma - 1)} \cdot \frac{p_{c_j^* i}}{B_i}, \quad (8)$$

where $y_i^{(j)}$ denotes y_i at the end of the process of user j.

All details of our algorithm are described in Algorithm 1. Recall that user j denotes the one who arrives at the jth round.

The following theorem is the main result in this section.

Theorem 2. *The competitive ratio of Algorithm 1 is $\alpha(1 - 1/\gamma)(1 - R_{\max})$.*

Notice that the competitive ratio approaches $\alpha(1 - 1/e^{1/\alpha})$ as $R_{\max}$ approaches 0. This together with Lemma 1 indicates that Algorithm 1 is a pseudo-polynomial time $(1 - 1/e)$-competitive algorithm for the problem in Section 3.1 under the small bid assumption. Since each round j takes $O(n^2 T_j)$ time and T_j is small (say, 30) in practice, the algorithm runs fast enough. We prove Theorem 2 by showing the following:

- $x = [x_{c,j}]_{j \in M, c \in \mathcal{C}_j}^\top$ computed by Algorithm 1 is feasible for (4),
- $y = \left[y_1^{(m)} \cdots y_n^{(m)} \right]^\top$, and $z = [z_1 \cdots z_m]^\top$ computed by Algorithm 1 constitute a feasible solution for (5),

Algorithm 1: Online algorithm for allocating outcomes

$y_i^{(0)} \leftarrow 0, \quad B_i^{(0)} \leftarrow B_i$ for all $i \in N$;
foreach $j \in M$ **do**
  $c_j^* \leftarrow$ an α-approx. sol. for $\max_{c \in \mathcal{C}_j} \sum_{i \in N} p_{ci}(1 - y_i^{(j-1)})$;
  define c'_j by $S_{c'_j} = \{i \in S_{c_j^*} \mid B_i^{(j-1)} \geq p_{c_j^* i}\}$;
  $p_{c'_j i} = 0$ if $i \in S_{c_j^*} \setminus S_{c'_j}$ and $p_{c'_j i} = p_{c^* i}$ otherwise;
  $x_{c'_j j} \leftarrow 1$ and $x_{cj} \leftarrow 0$ for all $c \in \mathcal{C}_j \setminus \{c'_j\}$;
  $B_i^{(j)} \leftarrow B_i^{(j-1)} - p_{c'_j i}$ for all $i \in N$;
  $z_j \leftarrow \sum_{i \in N} p_{c_j^* i}(1 - y_i^{(j-1)})/\alpha$;
  set $y_i^{(j)}$ by (8) for all $i \in N$;

- x, y, and z satisfy (6) with $\beta = \alpha(1 - 1/\gamma)(1 - R_{\max})$.

It is not difficult to prove the first two facts, and hence we focus on the proof for the last fact in this article due to the space constraint. First, we show that when advertiser i does not have an enough budget to pay the bid at some round, then the algorithm does not allocate i to subsequent users. The following lemma is proven by a similar proof to the one used in [Buchbinder *et al.*, 2007], and hence we omit the proof.

Lemma 2. *For each advertiser $i \in N$ and user $j \in M$, if $\sum_{j' \leq j} p_{c_{j'}^* i} > B_i$, then $y_i^{(j)} > 1$.*

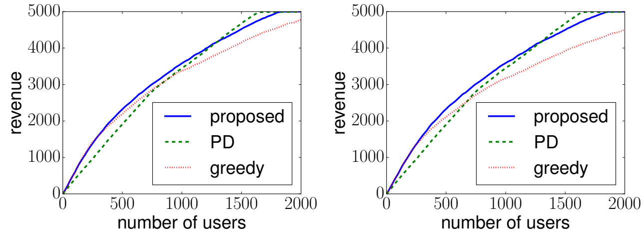
Since the algorithm avoids advertisers with little remaining budgets from $S_{c_j^*}$, the algorithm may miss chances to gain more revenue. However, the following lemma shows that such budgetary loss is not considerable compared to the revenue of the algorithm.

Lemma 3. *For each advertiser $i \in N$, it holds that $\sum_{j \in M} p_{c_j^* i} \leq \frac{1}{1 - R_{\max}} \sum_{j \in M} p_{c'_j i}$.*

Proof. If $\sum_{j \in M} p_{c_j^* i} \leq B_i$, then the statement holds. We assume the contrary. Let j^* denote the minimum index such that $\sum_{j \leq j^*} p_{c_j^* i} > B_i$. Lemma 2 implies that $y_i^{(j^*)} > 1$. For users $j > j^*$, we have $y_i^{(j)} \geq y_i^{(j^*)} > 1$, and hence $p_{c_j^* i} = 0$ by (7). Thus, it holds that $B_i \geq \sum_{j \leq j^*} p_{c_j^* i} - p_{c_{j^*}^* i} = \sum_{j \in M} p_{c'_j i} - p_{c_{j^*}^* i} \geq \left(1 - p_{c_{j^*}^* i}/B_i\right) \sum_{j \in M} p_{c'_j i} \geq (1 - R_{\max}) \sum_{j \in M} p_{c'_j i}$. ☐

Proof of Theorem 2. We denote by OPT and ALG the best revenue for the offline algorithms and the revenue for Algorithm 1, respectively. Let D_j be the objective values for the solution to (5) computed by Algorithm 1 at the end of the jth round—i.e., $D_j = \sum_{i \in N} B_i y_i^{(j)} + \sum_{j' \leq j} z_{j'}$. Since (y, z) is feasible for (5), OPT $\leq D_m$ holds.

We bound D_m by ALG. For $j = 0$, we have $D_0 = 0$. By the construction of $y_i^{(j)}$ and z_j at the jth round of the algorithm, $D_j - D_{j-1} = \sum_{i \in N} B_i(y_i^{(j)} - y_i^{(j-1)}) + z_j = \frac{\gamma}{\alpha(\gamma-1)} \sum_{i \in N} p_{c_j^* i}$ holds for each $j \geq 1$. Therefore, we have OPT $\leq D_m = \frac{\gamma}{\alpha(\gamma-1)} \sum_{i \in N} \sum_{j \in M} p_{c_j^* i} \leq \frac{\gamma}{\alpha(\gamma-1)(1-R_{\max})} \sum_{i \in N} \sum_{j \in M} p_{c'_j i} = \frac{\gamma}{\alpha(\gamma-1)(1-R_{\max})}$ALG, where the second inequality follows from Lemma 3. ☐



(a) Standard (b) Envy-free

Figure 1: A typical result for $n = 25$ and uniform budgets

4 Experiments

We evaluate performance of Algorithm 1 through computational experiments. The algorithm is compared with a greedy algorithm (called Greedy) and a primal-dual algorithm (called PD) obtained by modifying the one proposed by Buchbinder *et al.* [2007] for the ad-auction problem. These two algorithms compute an allocation to user j as follows:

Greedy: Find a set S of advertisers in $\mathrm{argmax}_{S \in \mathcal{S}_j} \sum_{i \in S} b_{ij}$, and allocate advertisers in S to user j.

PD: N_j is initialized to N. While $T_j \geq t_{ij}$ for some $i \in N_j$, allocate an advertiser $i \in \mathrm{argmax}_{i \in N_j} b_{ij}(1 - y_i)$ to j, and update $T_j \leftarrow T_j - t_i$ and $N_j \leftarrow N_j \setminus \{i\}$.

In the envy-free pricing setting, Greedy computes an allocation and a pricing by dynamic programming, ignoring the past budget consumptions of advertisers. PD computes an optimal pricing after deciding an allocation as above.

4.1 Results for Artificial Instances

First, we evaluate the performance of the algorithms over randomly generated instances. Each instance is generated as follows. We choose the number n of advertisers from 25, 50, and 100, and the number m of users from 500, 1000, and 2000. We assume two distributions of budgets: (1) uniform distribution ($B_i = 200$ for all $i \in N$), or (2) Pareto distribution (the minimum possible value is 100 and the mean is 200). Each bid b_{ij} is picked uniformly at random from $[0, 3]$. Thus, $R_{\max} \leq 3/100$. We assume that the length of each video-ad is same for any users, and hence the length of video-ad of advertiser i is written by t_i. Each t_i is generated uniformly at random from $[10, 45]$, and capacity T_j from $[10, 60]$.

Table 1 shows average revenues of each algorithm for each set of n, m, and the budget distribution. The average is taken over 100 instances for each parameter set. Here, "standard" means the standard setting of the video-ad allocation problem, and "envy-free" means the envy-free pricing setting of the problem. Figures 1a and 1b plot the revenue of each algorithm over the number of users for two typical random instances with $n = 25$ and uniform distribution.

As shown in Table 1, our algorithm outperforms the other two algorithms in most of the cases. The Greedy performs well when the number m of users is small. In this case, most

Table 1: Average revenues of the proposed algorithm, PD, and Greedy (100 trials)

			Standard			Envy-free		
	n	m	Proposed	PD	Greedy	Proposed	PD	Greedy
uniform distribution	25	500	**2280.4**	1829.3	2205.9	**2275.8**	1825.5	2117.2
		1000	**3606.2**	3485.3	3348.5	**3610.6**	3482.3	3184.6
		2000	4992.0	**4994.3**	4802.9	4980.0	**4989.2**	4570.3
	50	500	**2644.5**	1893.9	2615.2	**2639.0**	1894.4	2493.0
		1000	**4327.9**	3664.6	4102.4	**4325.7**	3665.2	3881.8
		2000	**6345.5**	6140.9	6022.0	**6349.0**	6139.8	5664.8
	100	500	3213.7	1948.2	**3280.9**	**3205.9**	1951.0	3117.0
		1000	**5458.8**	3821.1	5419.4	**5451.8**	3823.5	5097.2
		2000	**8861.8**	7445.6	8413.4	**8854.4**	7448.9	7792.7
Pareto distribution	25	500	**2195.8**	1851.5	2123.1	**2190.8**	1849.5	2041.5
		1000	**3480.2**	3423.9	3245.5	**3479.2**	3420.0	3112.1
		2000	4700.2	**4769.6**	4533.6	4689.8	**4773.4**	4391.1
	50	500	**2790.0**	1939.1	2775.4	**2783.0**	1940.4	2661.6
		1000	**4682.2**	3779.1	4491.5	**4678.5**	3779.5	4300.4
		2000	**7323.1**	7087.5	6787.2	**7328.0**	7082.6	6433.2
	100	500	3372.9	1959.7	**3422.2**	**3366.6**	1962.3	3279.9
		1000	5822.4	3856.1	**5827.2**	**5817.2**	3858.6	5534.3
		2000	**9708.9**	7583.8	9326.5	**9704.8**	7585.7	8744.6

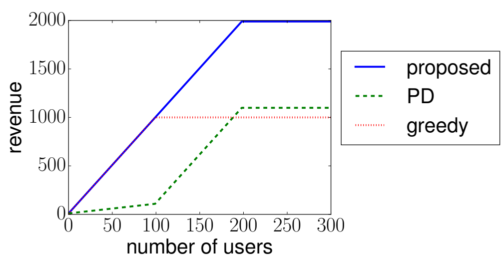


Figure 2: Results over a carefully constructed instances

advertisers still have enough budgets at the end. It is also observed from Figures 1a and 1b that the performance of our algorithm is same as the Greedy algorithm when m is small. However, the Greedy algorithm falls behind as budgets of advertisers are exhausted. The performance of the PD algorithm is worst when m is small, and it is still worse than our algorithm as a whole. When m is large, level-offs occurred. In this situation, almost all advertisers exhausted the budgets.

To demonstrate the superiority of our algorithm, we also compare the algorithms on a carefully constructed instance. In this instance, parameters are set so that $n = 1110$, $m = 300$, and $R_{\max} = 0.01$. All users have capacity 10. The user set is divided into U_1 and U_2 so that $|U_1| = 100$ and $|U_2| = 200$. Users in U_1 arrive first, and then the others arrive. The budget of each advertiser is 100, and the advertisers are divided into three groups A_1, A_2, A_3. The first group A_1 consists of 10 advertisers with $t_i = 1$ and $b_{ij} = 1$ for all users j. The second group A_2 consists of 1000 advertisers with $t_i = 1$, $b_{ij} = 1 - \varepsilon$ for $j \in U_1$, and $b_{ij} = 0$ for the others. The third group A_3 consists of 100 advertisers with $t_i = 10$, $b_{ij} = 1$ for $j \in U_1$, and $b_{ij} = 0$ for the others.

On this instance, results do not change for the standard and the envy-free settings. Figure 2 shows the revenues of the algorithms. In this instance, the Greedy algorithm spends all budgets of advertisers in A_1 for users in U_1, and it obtains no revenue after that. The PD algorithm obtains revenue 1 per user in U_1, while our algorithm gains revenue at least $10(1 - \varepsilon)$ per user. Observe that the final revenues of the algorithms are significantly different in this instance.

Table 2: Budget consumption rate (%) and total computational time for an instance constructed from a real data set

	Proposed	PD	Greedy	total time (s)
Standard	**73.6**	72.2	72.6	1224.4
Envy-free	**73.5**	71.7	70.7	2271.0

4.2 Results for Instances with a Real Dataset

In this subsection, we report results on an instance constructed from a real dataset. The dataset consists of browsing records of video-ads in a streaming site operated by Yahoo! JAPAN. Each record contains information on which users (distinguished by browser cookie) watched which ads. The dataset also includes the length of each video-ad, an approximate value of each budget, and viewing time of each user. There were 82 ads and 21,306,810 queries in our dataset. The streaming service sometimes allocated two ads to a query from a user. Such cases occurred on about 10% out of queries. From this dataset, we construct a problem instance as follows. We construct a set of $n = 82$ advertisers and a set of $m = 21{,}306{,}810$ users. We set the budget B_i for $i \in N$ and the viewing time T_j for $j \in M$ according to the dataset. The average of T_j is 18 seconds. Each bid from advertisers with no user preference is set to 2. If an advertiser input user preference, its bid for users satisfying the preference is set to an integer chosen randomly from $[2, 10]$. We also set the video length t_{ij} for advertiser i and user j from the user preference input by the advertisers and the browsing history of users, which we omit the details due to space limitation.

The results are described in Table 2. Here, each cell denotes the ratio (%) of revenue to total budget of all advertisers or total computational time. We can observe that the revenues of the proposed algorithm are better than the other algorithms by more than 1%. Moreover, the total computational times are short enough as it takes about 0.1ms per user.

Let us summarize experimental results. First, the Greedy algorithm performs best in the beginning of inputs, where no advertiser exhausts the budget. Our algorithm performs as well as the Greedy even in the beginning, and does better at the end. The PD algorithm gains more revenues than the Greedy at the end. However, since the PD is based on a greedy method for the knapsack problem, its revenue is sometimes significantly small compared with our algorithm. Our algorithm outperforms the others in most cases, including real-data instances. In addition, our algorithm can respond to each user quickly enough in practical settings. Therefore, our algorithm is practically useful.

5 Conclusion

In this paper, we formulated a new optimization problem, which models an important step of video-ad advertising, and presented an online algorithm with theoretical performance guarantee for it. There are many interesting directions of future studies on this topic. One direction is to predict users' viewing capacities. Our algorithms suppose that the viewing capacity of each user is revealed upon the user's arrival. For user experiences and effective advertisement, it is an important future work to develop a method for this estimation.

References

[Balseiro *et al.*, 2014] Santiago R. Balseiro, Jon Feldman, Vahab Mirrokni, and S. Muthukrishnan. Yield optimization of display advertising with ad exchange. *Management Science*, 60(12):2886–2907, 2014.

[Bateni *et al.*, 2014] MohammadHossein Bateni, Jon Feldman, Vahab Mirrokni, and Sam Chiu-wai Wong. Multiplicative bidding in online advertising. In *Proceedings of the 15th ACM Conference on Economics and Computation (EC '14)*, pages 715–732, 2014.

[Bhalgat *et al.*, 2012] Anand Bhalgat, Jon Feldman, and Vahab Mirrokni. Online allocation of display ads with smooth delivery. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pages 1213–1221, 2012.

[Bhalgat *et al.*, 2014] Anand Bhalgat, Nitish Korula, Hannadiy Leontyev, Max Lin, and Vahab Mirrokni. Partner tiering in display advertising. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14)*, pages 133–142, 2014.

[Bharadwaj *et al.*, 2012] Vijay Bharadwaj, Peiji Chen, Wenjing Ma, Chandrashekhar Nagarajan, John Tomlin, Sergei Vassilvitskii, Erik Vee, and Jian Yang. SHALE: an efficient algorithm for allocation of guaranteed display advertising. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pages 1195–1203, 2012.

[Buchbinder *et al.*, 2007] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Conference on Algorithms (ESA '07)*, pages 253–264, 2007.

[Goel *et al.*, 2010] Ashish Goel, Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Advertisement allocation for generalized second-pricing schemes. *Operations Research Letters*, 38(6):571–576, 2010.

[Hojjat *et al.*, 2014] Ali Hojjat, John Turner, Suleyman Cetintas, and Jian Yang. Delivering guaranteed display ads under reach and frequency requirements. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI '14)*, pages 2278–2284, 2014.

[Ieong *et al.*, 2014] Samuel Ieong, Mohammad Mahdian, and Sergei Vassilvitskii. Advertising in a stream. In *Proceedings of the 23rd International World Wide Web Conference (WWW '14)*, pages 29–38, 2014.

[Interactive Advertising Bureau, 2015] Interactive Advertising Bureau. IAB internet advertising revenue report, 2014 full year results. http://www.iab.net/media/file/IAB_Internet_Advertising_Revenue_FY_2014.pdf, 2015.

[Mehta *et al.*, 2007] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5), 2007.

[Pew Research Center, 2014] Pew Research Center. State of the news media 2014: News video on the web: A growing, if uncertain, part of news. http://www.journalism.org/files/2014/03/News-Video-on-the-Web.pdf, 2014.

[Radovanovic and Heavlin, 2012] Ana Radovanovic and William D. Heavlin. Risk-aware revenue maximization in display advertising. In *Proceedings of the 21st World Wide Web Conference (WWW '12)*, pages 91–100, 2012.