Applied Soft Computing Journal 93 (2020) 106370

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Online advertising assignment problem without free disposal

Gwang Kim [a], Ilkyeong Moon [a,b,*]

[a] Department of Industrial Engineering, Seoul National University, Seoul, Republic of Korea
[b] Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 20 August 2019
Received in revised form 1 April 2020
Accepted 28 April 2020
Available online 19 May 2020

Keywords:
Online bipartite matching
Online algorithms
Deterministic algorithm
Stochastic programming
Benders decomposition

ABSTRACT

This paper presents an online advertising assignment problem that generalizes the online version of the bipartite matching problem. Specifically, it focuses on the *Display Ads* problem, which is a generalization of the edge-weighted and capacitated matching problem. The display ads problem has been studied alongside the property of *free disposal*, in which an advertisement is allowed to be matched more times than its capacity. Although the problem with free disposal is tractable, the problem situation might be restricted and challenging to apply to other types of problems. The objective of this research on the display ads problem is to maximize the total weight of matched edges while considering a strict capacity constraint. This paper analyzes two online input orders (adversarial and probabilistic orders) to the problem. For the adversarial order, we design deterministic algorithms with worst-case guarantees and prove the competitive ratios of them. Upper bounds for the problem are also proposed. For the probabilistic order, stochastic online algorithms, consisting of scenario-based stochastic programming and Benders decomposition, are presented. We conduct numerical experiments of the stochastic online algorithm in two probabilistic order models (known IID and random permutation).

1. Introduction

With an increase in online users, online advertising has become a significant source of income for many Internet-based companies. According to an Internet Advertising Bureau (IAB) report [1], Internet advertising revenue in the United States was $88.0 billion in 2017. The main advantages of online advertising are traceability, cost-effectiveness, reach, and interactivity. They facilitate the continuous popularity of online advertising [2]. For these reasons, managers who place online advertisements on their websites need to develop decision-making processes to maximize revenue. One such process involves rapidly selecting an appropriate advertisement from among those in a set of available advertisements, and then assigning it on the website, a placement defined as "a slot" [3].

In this study, an online advertising assignment problem that managers solve corresponds to a bipartite matching problem in graph theory. The problem can be interpreted as finding an optimal matching among the ads and the slots because the assignment of advertisements is generally decided either by auction or through contracts [4]. Unlike with a matching problem, in which cardinality is maximized, the objective of the online advertising assignment problem is to find connections, through which revenue is maximized, between the ads and the slots. Each edge (connection) has a weight. The weight of the edge might be a prediction of click-through probability, an estimate of targeting quality, or a bid submitted by the advertiser [5,6]. When an advertisement is assigned to a slot, the weight corresponding to the edge is realized.

In reality, we do not know the information on weights beforehand, making the problem uncertain. Because of this, managers focus on the online version of the problem [7–9]. In other words, we define a bipartite graph for which information about the nodes on the left-hand side is known in advance, and for which the nodes on the right-hand side arrive online (one node at a time). The nodes on the left-hand side represent ads, and those on the right-hand side represent slots. When a node on the right-hand side arrives, the edges and weights incident on the node are revealed. An online algorithm of the problem selects one of the edges (an ad is displayed on the slot) or discards them (no ad is displayed on the slot). The decision is irrevocable [10–26]. Online algorithms must complete each request of assignment without knowing the future sequence of the nodes on the right-hand side [27].

If each node on the left-hand side has an integer capacity (the maximum number of being matched to the nodes on the right-hand side), we call the situation *Display Ads* problem. The

* Corresponding author at: Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea.
E-mail address: ikmoon@snu.ac.kr (I. Moon).

https://doi.org/10.1016/j.asoc.2020.106370
1568-4946/

problem is a generalization of the edge-weighted and capacitated online bipartite matching problem. Feldman et al. [6] gave a $(1-\frac{1}{e})$-competitive algorithm for the display ads problem in the adversarial order when the value of each capacity is big. Korula and Pál [28] developed a $\frac{1}{8}$-approximation algorithm for the online weighted bipartite matching problem in the random order. Kesselheim et al. [29] showed better result by providing an $\frac{1}{e}$-approximation algorithm in the random order. Feldman et al. [30] and Bhalgat et al. [31] implemented fairness and smooth delivery constraints for the display ads problem. For edge-weighted bipartite matching [32], proposed a near-optimal algorithm for the edge-weighted b-matching problem. Sun et al. [33] proposed a randomized algorithm, which gives a near-optimal solution.

In the display ads problem Feldman et al. [6], introduced the property (assumption) referred to as *free disposal*. The definition of the free disposal assumption is that each node on the left-hand side is allowed to be matched more times than its capacity, (c), but the managers gain only for the "c" highest weights matched. In other words, the assumption allows for violating the capacity constraint. Previous research introduced the assumption in the display ads problem to obtain bounded competitive ratios in the adversarial order [6,30,31]. Although the problem is tractable when the assumption is allowed, the problem situation might be restricted. If some advertisers are sensitive to the number of times their advertisements are displayed, the solutions with the assumption might cause issues with trust. That is because there is a possibility that an ad can be displayed more times than its capacity, while other ads miss the chance to be displayed. Also, the display ads problem that allows for the free disposal assumption is challenging to apply to other types of problems (e.g., scheduling or resource allocation) in which resources, such as humans and machines, are strictly limited.

The *Adwords* problem is similar to the display ads problem in terms of application of online bipartite matching problems. The adwords problem has an individual budget instead of an integer capacity for each node on the left-hand side and the objective of the problem is to maximize the total budget spent [16,34–44]. Bhaskar et al. [35] proposed the adwords problem without a small-bid assumption, which is a relaxed capacity constraint. Like Bhaskar et al. [35], we propose a display ads problem that does not allow for the free disposal assumption. We call our problem "the display ads problem without free disposal". The objective of this problem is to maximize the total weight of edges matched, while considering the strict capacity constraint.

This paper presents the analyses of two online input orders (adversarial and probabilistic orders) to the problem. For the adversarial order, we design deterministic algorithms with worst-case guarantees and prove the competitive ratios of them. Upper bounds for the problem also are proposed. For the probabilistic order, information on the probability distribution of the future coming slots is known. In real-life situations, the company can stochastically estimate the input sequence by using historical data. We consider two probabilistic order models (known IID and random permutation) in this paper. The two probabilistic orders are generally used as stochastic input models of the online matching problem [4].

This paper presents stochastic online algorithms with scenario-based stochastic programming and Benders decomposition. The stochastic online algorithm is based on the algorithms presented in Feldman et al. [6] and Legrain and Jaillet [42]. Upon each arrival of a node on the right-hand side, the algorithm estimates future scenarios of remaining slots and solve optimization problems to make a good decision for the current slot. The algorithm is used in this study to handle the uncertainty reasonably. Numerical experiments on various future scenarios are conducted to show improved performances over the algorithm of Feldman et al. [6].

The former (adversarial) is a theoretical perspective, while the latter (probabilistic) is a practical perspective. Table 1 shows the comparison between this study and recent research on the display ads problem. The literature in Table 1 is categorized by equal/unequal edge weight, 1/N(with free disposal)/N(without free disposal) capacity, and theoretical/practical approaches.

2. Display ads problem without free disposal

The display ads problem without free disposal is defined as follows [4,6,32,33]. For an edge-weighted bipartite graph $\boldsymbol{G} = (A, T, E, w)$, A is a set of advertisements (left); T is a set of slots (right); E is a set of edges of graph $\boldsymbol{G}$; and w is a set of weights for E. We know the information on A in advance and each advertisement $i \in A$ has capacity C_i, which is the maximum number of being matched to T. However, we do not know any information of T, E, and w, except $|T|$. The set of T arrives online, one node at a time. When a node $j \in T$ arrives, all edges incident to j as well as the weights, w_{ij} of each are revealed. The algorithm matches a connection between a node j and one of the advertisements available or leaves the node unmatched. The decision made is irrevocable. To resolve the absence of non-trivial competitive ratios, we assume that the online algorithms know the range of the weights $[L_i, U_i]$ for each advertisement i.

A mathematical formulation for the display ads problem is as follows:

$$\max \sum_{i=1}^{|A|}\sum_{j=1}^{|T|} w_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_{ij} \le 1 \ \forall j \in T \tag{2}$$

$$\sum_{j=1}^{|T|} x_{ij} \le C_i \ \forall i \in A \tag{3}$$

$$x_{ij} \in \{0, 1\} \ \forall i \in A, j \in T \tag{4}$$

The binary decision variable, x_{ij} is 1 if advertisement $i \in A$ is matched to slot $j \in T$; 0 otherwise. The objective function (1) maximizes the total weight of the edges matched while satisfying the capacity constraint for each advertisement. Constraint (2) ensures that a slot can display at most one advertisement. Constraint (3) limits the number of times each advertisement can be displayed.

This paper focuses on the online version of the problem and proposes online algorithms to solve the problem. This paper deals with two online input orders: adversarial and probabilistic. We assume that there is no knowledge of the arrival order of T in the adversarial order, but there is the arrival order of T under the probabilistic structure in the probabilistic order. For the sake of simplicity and tractability, the weight range for each advertisement i ($w_{ij} \in [L_i, U_i]$, $\forall j$ adjacent to i) is assumed to be known in advance. If the online algorithm finds a matching M for graph $\boldsymbol{G}$ then the objective value of the algorithm is $\sum_{(i,j)\in M} w_{ij}$. We use the notation of "competitive ratio" to measure the performance of the online algorithm. The competitive ratio is defined as the ratio of the value obtained by the online algorithm (ALG) to the optimal offline objective value (OPT) given a bipartite graph $\boldsymbol{G}$. For every graph $\boldsymbol{G} = (A, T, E, w)$ and every order of T, the online algorithm is c-competitive if $ALG \ge c \cdot OPT$.

The remainder of the paper is organized as follows. Section 3 presents deterministic algorithms and some theorems for the adversarial order. Section 4 introduces stochastic online algorithms with scenario-based stochastic programming and Benders decomposition for the probabilistic order. Section 5 provides numerical results of the stochastic online algorithm that is presented in Section 4. Section 6 offers contributions and conclusions of this study.

Table 1
Comparison between recent research and this study.

Author (year)	Edge-weighted	Capacity	Approach
Feldman et al. (2009) [6]	Unequal	N(w[a])	T[c]
Korula and Pál (2009) [28]	Unequal	1	T
Haeupler et al. (2011) [17]	Unequal	1	T/P[d]
Bhalgat et al. (2012) [31]	Unequal	N(w)	T/P
Kesselheim et al. (2013) [29]	Unequal	1	T
Jaillet and Lu (2014) [19]	Equal	1	T/P
Chen and Wang (2015) [13]	Unequal	1	T/P
Ting and Xiang (2015) [32]	Unequal	N(w/o[b])	T
Bhaskar et al. (2016) [35]	Unequal	N(w/o)	T
Sun et al. (2017) [33]	Unequal	1	T
Huang et al. (2018) [18]	Unequal	1	T/P
This Paper	Unequal	N(w/o)	T/P

[a]w indicates 'with free disposal assumption.'
[b]w/o indicates 'without free disposal assumption.'
[c]T indicates a theoretical approach.
[d]P indicates a practical approach.

3. Deterministic algorithms for adversarial order

In this section, deterministic algorithms are described for the display ads problem without free disposal. We have no knowledge of the arrival order of T over any bipartite graph $\boldsymbol{G} = (A, T, E, w)$ We assume that the range of the weights for an advertisement $i \in A$ is $[L_i, U_i]$ and the capacity of it is C_i. A simple deterministic algorithm, called *Greedy*, is defined as follows:

Algorithm 1: Greedy

while *a new node $j \in T$ arrives* **do**
  if *all neighbors of j are unavailable (not or full connected)* **then**
    continue;
  else
    match j to that available neighbor i which has the maximum value of w_{ij};
  end
end

For Algorithm 1, we prove the following competitive ratio:

Theorem 1. *Algorithm 1 has a competitive ratio of $\frac{1}{1+M_1}$ for the display ads problem without free disposal ($M_1 := \max(\frac{U_i}{L_i}), \forall i \in A$). (See proof in Appendix A.1.)*

For this problem, we consider a worst case in which the competitive ratio of any deterministic algorithms could be affected: For example, for an advertisement $i \in A$, a deterministic algorithm has already matched i to C_i nodes in T. All the edges matched have low weights. Then, a node in T, which is the only neighbor of i, arrives. The weight of the edge between them is extremely high. To avoid this case, we propose Algorithm 2 which is based on the techniques developed by Ting and Xiang [32]. For Algorithm 2, we define variables x_i, as the number of matched edges between i and the nodes in T.

For Algorithm 2, we prove the following competitive ratio as follows:

Theorem 2. *Algorithm 2 has a competitive ratio of $\frac{1}{1+M_2}$ for the display ads problem without free disposal ($M_2 := \max\left(C_i \cdot \lfloor\frac{C_i}{k_i}\rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}}\right)$ such that $k_i := \min(C_i, \lceil \ln\frac{U_i}{L_i}\rceil)$, $\forall i \in A$). (See proof in Appendix A.2.)*

In this paper, we propose an integrated algorithm that combines Algorithm 1 and Algorithm 2, and prove the following lemma:

Algorithm 2: Greedy with sub_ads

for *each $i \in A$* **do**
  if $L_i = U_i$ **then**
    $k_i \leftarrow 1$;
  else
    $k_i \leftarrow \min(C_i, \lceil \ln\frac{U_i}{L_i}\rceil)$
  end
  Decompose a variable x_i into k_i variables $x_{i0}, x_{i1}, \ldots, x_{i(k_i-1)}$ and set all variables to 0;
end
while *a new node $j \in T$ arrives* **do**
  $t \leftarrow 1$;
  while $t \leq |A|$ **do**
    $i \leftarrow$ a neighbor of j such that the weight of edge between them is the t^{th} highest among that of all edges adjacent to j;
    if $i = \emptyset$ **then**
      break;
    end
    Find an integer value p such that $w_{ij} \in [L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}, L_i(\frac{U_i}{L_i})^{\frac{p+1}{k_i}})$;
    if $x_{ip} < \lfloor\frac{C_i}{k_i}\rfloor$ **then**
      match j to i and $x_{ip} \leftarrow x_{ip} + 1$, break;
    else
      $t \leftarrow t + 1$;
    end
  end
end

Algorithm 3: Greedy + Greedy with sub_ads

if $M_1 \leq M_2$ **then**
  run Algorithm 1;
else
  run Algorithm 2;
end

$M_1 := \max(\frac{U_i}{L_i})$ and $M_2 := \max\left(C_i \cdot \lfloor\frac{C_i}{k_i}\rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}}\right)$ such that $k_i := \min(C_i, \lceil \ln\frac{U_i}{L_i}\rceil), \forall i \in A$

Lemma 1. *Algorithm 3 has a competitive ratio of* $\max\left(\frac{1}{1+M_1}, \frac{1}{1+M_2}\right)$ *for the display ads problem without free disposal.*

Feldman et al. [6] presented a simple upper bound for the problem.

Theorem 3. *No deterministic algorithm for the display ads problem achieves a competitive ratio better than $\frac{1}{2}$ even though free disposal is allowed.*

Proof. We consider an instance from $\boldsymbol{G} = (A, T, E, w)$ in which $A = \{i_1, i_2\}$ and $T = \{j_1, j_2\}$. The capacity of each advertisement is 1. When j_1 arrives first, we know that $w_{i_1 j_1}$ and $w_{i_2 j_1}$ have the same weight (let the weight be w). Once j_1 has been matched, j_2 arrives. The edge incident to the same advertisement matched to j_1 is revealed only and the weight of the edge is w. As j_2 cannot be matched when it arrives, a deterministic algorithm can obtain w. However, the optimal objective value is $2w$. □

Using Theorem 3, we give the following lemma:

Lemma 2. *Algorithm 3 can be an optimal algorithm when $L_i = U_i, \ \forall i$.*

Proof. When $L_i = U_i$ for every advertisement $i \in A$, the values of M_1 and M_2 in Algorithm 3 are all equal to 1. Hence, the competitive ratio of the algorithm can be $\frac{1}{2}$. As the upper bound of the competitive ratio is $\frac{1}{2}$ according to Theorem 3, Algorithm 3 can be an optimal algorithm when $L_i = U_i, \ \forall i$. □

To show that Algorithm 3 features a good competitive ratio even when $L_i \neq U_i$, we prove an upper bound on the competitive ratio for any deterministic algorithm and show that the gap between the upper bound and competitive ratio is not large. We present the following theorem, which is based on the techniques developed by Ting and Xiang [32], to get an upper bound.

Theorem 4. *No deterministic algorithm for the display ads problem without free disposal can have a competitive ratio larger than $\min(\frac{1}{2}, K_1, K_2)$. $K_1 := \min\Big(\frac{1}{(\frac{U_i}{L_i})^{\frac{1}{C_i}}}\Big)$ and $K_2 := \min\Big(\frac{1}{\lceil \ln \frac{U_i}{L_i} \rceil} \cdot \frac{e}{e-1}\Big), \ \forall i \in A$. (See proof in Appendix A.3.)*

When the ratio of L_i and U_i is very small, the upper bound and lower bound each approaches $\frac{1}{2}$. Otherwise, we divide the bounds into two cases according to the relative size of C_i compared to the ratio of L_i and U_i. When C_i is small, the upper bound approaches K_1 and the lower bound does $\frac{1}{\left(1+(\frac{U_i}{L_i})^{\frac{1}{C_i}}\right)}$. When C_i is large, the upper bound approaches K_2 and the lower bound does $\frac{1}{\left(1+\lceil \ln \frac{U_i}{L_i} \rceil\right)}$. The gaps between the upper and lower bounds are not large when the ratio of L_i and U_i is close to 1 or C_i is very large or small compared to the ratio of L_i and U_i. It implies that the deterministic algorithm presented shows good performances in the adversarial order. When it comes to randomized algorithms, Sun et al. [33] proposed a near-optimal algorithm for the problem. For this reason, we do not cover randomized algorithms in this paper.

4. Stochastic algorithms for probabilistic order

In this section, we assume that the managers already know information on the probability distribution for the future coming slots. Stochastic online algorithms are proposed for the display ads problem in the probabilistic order. The algorithm combines the primal–dual algorithms developed by Feldman et al. [6] with scenario-based stochastic programming and Benders decomposition proposed by Legrain and Jaillet [42]. The algorithm in the probabilistic order is a practical approach. Numerical experiments of stochastic processes are conducted, and the results are presented in Section 5. We uses two stochastic models according to information of the sequence on the right-hand nodes: the *known IID* and *random permutation* models.

4.1. Known IID model

For the known IID model, we assume that there are some types of nodes in T. For a collection K of node types, the managers know a probability distribution on K in advance. For each slot, a type node $k \in K$ is drawn from the probability distribution. We suppose that the jth slot has just arrived (such that k_j, which is a node type at slot j, is revealed) and that the managers decide an advertisement to be displayed on the slot. The parameters and decision variables for stochastic programming formulation are introduced in Table 2.

A stochastic programming formulation at the time of the jth slot is as follows:

$$\max \sum_i w_{ik_j} x_i + \sum_{\omega \in \Omega_j} p^w \sum_i \sum_k w_{ik} y_{ik}^{\omega} \tag{5}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{6}$$

$$\sum_{i=1}^{|A|} y_{ik}^{\omega} \leq T_{jk}^{\omega} \quad \forall \omega \in \Omega_j, k \in K \tag{7}$$

$$x_i + \sum_k y_{ik}^{\omega} \leq C_i^{left} \quad \forall \omega \in \Omega_j, i \in A \tag{8}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{9}$$

$$y_{ik}^{\omega} \in \mathbb{N} \quad \forall i \in A, k \in K, \omega \in \Omega_j \tag{10}$$

The objective function (5) maximizes the weight for the jth slot and the expected total weight obtained from the remaining future slots. Constraint (6) ensures that the jth slot can display at most one advertisement. Constraint (7) guarantees that the number of slots for type k allocated to all advertisements in scenario ω cannot exceed T_{jk}^{ω}. Constraint (8) limits the capacity left for each advertisement and each scenario. Constraints (9) and (10) define x_i and y_{ik}^{ω} as binary and integer variables, respectively.

The stochastic formulation can be decomposed using Benders decomposition [45]:

Master problem

$$\max \sum_i w_{ik_j} x_i + \sum_{\omega \in \Omega_j} p^{\omega} S(\boldsymbol{x}, \omega) \tag{11}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{12}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{13}$$

Slave problems (for each $\boldsymbol{x}$ and ω)

$$S(\boldsymbol{x}, \omega) = \max \sum_i \sum_k w_{ik} y_{ik}^{\omega} \tag{14}$$

$$\text{s.t.} \sum_{i=1}^{|A|} y_{ik}^{\omega} \leq T_{jk}^{\omega} \quad \forall k \in K \tag{15}$$

$$\sum_k y_{ik}^{\omega} \leq C_i^{left} - x_i \quad \forall i \in A \tag{16}$$

$$y_{ik}^{\omega} \in \mathbb{N} \quad \forall i \in A, k \in K \tag{17}$$

For each $\boldsymbol{x}$ and ω, the objective value of the slave problem can be calculated to solve the problem. The value can be approximated by using the dual of the slave problem. α_k^{ω} and β_i^{ω} are dual variables corresponding to the first and second type of constraint for each slave problem, respectively. Using the weak duality theorem, we show that $S(\boldsymbol{x}, \omega) \leq \sum_k T_{jk}^{\omega} \alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega}$ for every $\boldsymbol{x}$ and ω. The objective value of the dual problem

Table 2
Parameters and decision variables (known IID)

Ω_j	Set of future sample scenarios at $j \in T$
p^{ω}	Probability of a scenario $\omega \in \Omega_j$
w_{ik}	Weight of the edge between $i \in A$ and $k \in K$
T^{ω}_{jk}	Number of slots for type k in scenario ω
C^{left}_i	Capacity left for an advertisement i
x_i	Binary decision variable, whose value is 1 if ad i is allocated to slot j, 0 otherwise
y^{ω}_{ik}	Number of slots for type k allocated to advertisement i for scenario ω (integer variable)

can be a cut for the master problem:
Dual slave problems

$$\min \sum_k T^{\omega}_{jk}\alpha^{\omega}_k + \sum_i (C^{left}_i - x_i)\beta^{\omega}_i \tag{18}$$

$$\text{s.t. } \alpha^{\omega}_k + \beta^{\omega}_i \geq w_{ik} \quad \forall i \in A, k \in K \tag{19}$$

$$\alpha^{\omega}_k \geq 0, \beta^{\omega}_i \geq 0, \quad \forall i \in A, k \in K \tag{20}$$

Master problem with an added cut

$$\max \sum_i w_{ik_j}x_i + \sum_{\omega\in\Omega_j} p^{\omega}S^{\omega} \tag{21}$$

$$\text{s.t. } \sum_{i=1}^{|A|} x_i \leq 1 \tag{22}$$

$$S^{\omega} \leq \sum_k T^{\omega}_{jk}\alpha^{\omega}_k + \sum_i (C^{left}_i - x_i)\beta^{\omega}_i \quad \forall \omega \in \Omega_j \tag{23}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{24}$$

Because the master problem with an added cut is a maximization problem, the inequality $S^{\omega} \leq \sum_k T^{\omega}_{jk}\alpha^{\omega}_k + \sum_i (C^{left}_i - x_i)\beta^{\omega}_i$ becomes equality. Accordingly, the objective function of the master problem is replaced by $\sum_i w_{ik_j}x_i + \sum_{\omega\in\Omega_j} p^{\omega}[\sum_k T^{\omega}_{jk}\alpha^{\omega}_k + \sum_i (C^{left}_i - x_i)\beta^{\omega}_i]$. If the constants are eliminated, then the objective function can be $\sum_i w_{ik_j}x_i - \sum_{\omega\in\Omega_j} p^{\omega}[\sum_i \beta^{\omega}_i x_i]$. By using the value of the dual variables β^{ω}_i, we develop a stochastic online algorithm with the primal–dual algorithm, which is presented in Section 4.3.

4.2. Random permutation model

For the random permutation model, we assume that there are $|T|$ node types in T and the number of slots for each type is 1. For a collection K of node types, it becomes $K = |T|$ and $T = \{1, 2, \ldots, |T|\}$. That is, a sequence on the right-hand side nodes becomes one of the random permutations of T. The probability of each sequence is identical. Like with the known IID model, we suppose that the jth slot has just arrived (such that a type at slot j is revealed) and that the managers choose an advertisement to be displayed on the slot. The parameters and decision variables for stochastic programming formulation are introduced in Table 3.

A stochastic programming formulation at the time of the jth slot is as follows:

$$\max \sum_i w_{ij}x_i + \sum_{\omega\in\Omega_j} p_j \sum_i \sum_{k\geq j+1} w^{\omega}_{ik}x^{\omega}_{ik} \tag{25}$$

$$\text{s.t. } \sum_{i=1}^{|A|} x_i \leq 1 \tag{26}$$

$$\sum_{i=1}^{|A|} x^{\omega}_{ik} \leq 1 \quad \forall \omega \in \Omega_j, k \geq j+1 \tag{27}$$

$$x_i + \sum_{k\geq j+1} x^{\omega}_{ik} \leq C^{left}_i \quad \forall \omega \in \Omega_j, i \in A \tag{28}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{29}$$

$$x^{\omega}_{ik} \in \mathbb{B} \quad \forall i \in A, k \geq j+1, \omega \in \Omega_j \tag{30}$$

The objective function (25) maximizes the weight for the jth slot and the expected total weight obtained by the remaining future slots. Constraints (26) and (27) ensure that each slot of each scenario can display at most one advertisement. Constraint (28) limits the capacity left for each advertisement and each scenario. Constraints (29) and (30) define x_i and x^{ω}_{ik} as binary variables, respectively. Like the known IID model, the stochastic formulation can be decomposed by using Benders decomposition [45]:
Master problem

$$\max \sum_i w_{ij}x_i + \sum_{\omega\in\Omega_j} p_j S(\boldsymbol{x}, \omega) \tag{31}$$

$$\text{s.t. } \sum_{i=1}^{|A|} x_i \leq 1 \tag{32}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{33}$$

Slave problems (for each $\boldsymbol{x}$ and ω)

$$S(\boldsymbol{x}, \omega) = \max \sum_i \sum_k w^{\omega}_{ik}x^{\omega}_{ik} \tag{34}$$

$$\text{s.t. } \sum_{i=1}^{|A|} x^{\omega}_{ik} \leq 1 \quad \forall k \geq j+1 \tag{35}$$

$$\sum_{k\geq j+1} x^{\omega}_{ik} \leq C^{left}_i - x_i \quad \forall i \in A \tag{36}$$

$$x^{\omega}_{ik} \in \mathbb{B} \quad \forall i \in A, k \geq j+1 \tag{37}$$

For each $\boldsymbol{x}$ and ω, the objective value of the slave problem can be calculated to solve the problem. The value can be approximated by using the dual of the slave problem. α^{ω}_k and β^{ω}_i are dual variables corresponding to the first and second type of constraint for each slave problem, respectively. Using the weak duality theorem, we show that $S(\boldsymbol{x}, \omega) \leq \sum_k T^{\omega}_{jk}\alpha^{\omega}_k + \sum_i (C^{left}_i - x_i)\beta^{\omega}_i$ for every $\boldsymbol{x}$ and ω. The objective value of the dual problem can be a cut for the master problem:
Dual slave problems

$$\min \sum_k \alpha^{\omega}_k + \sum_i (C^{left}_i - x_i)\beta^{\omega}_i \tag{38}$$

$$\text{s.t. } \alpha^{\omega}_k + \beta^{\omega}_i \geq w_{ik} \quad \forall i \in A, k \geq j+1 \tag{39}$$

$$\alpha^{\omega}_k \geq 0, \beta^{\omega}_i \geq 0, \quad \forall i \in A, k \geq j+1 \tag{40}$$

Master problem with an added cut

$$\max \sum_i w_{ij}x_i + \sum_{\omega\in\Omega_j} p_j S^{\omega} \tag{41}$$

$$\text{s.t. } \sum_{i=1}^{|A|} x_i \leq 1 \tag{42}$$

Table 3
Parameters and decision variables (random permutation)

Ω_j	Set of future sample scenarios at $j \in T$, $\lvert\Omega_j\rvert = (\lvert T\rvert - j)!$
p_j	Probability of each scenario at $j \in T$, $p_j = \frac{1}{\lvert\Omega_j\rvert}$
w_{ik}^{ω}	Weight of the edge between $i \in A$ and $k \in K$ for scenario ω ($k \geq j+1$)
C_i^{left}	Capacity left for an advertisement i
x_i	Binary decision variable, whose value is 1 if ad i is allocated to slot j, 0 otherwise
x_{ik}^{ω}	Binary decision variable, whose value is 1 if advertisement i is allocated to slot k in the scenario ω; 0 otherwise ($k \geq j+1$)

$$S^{\omega} \leq \sum_{k \geq j+1} \alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega} \quad \forall \omega \in \Omega_j \tag{43}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{44}$$

Because the master problem with an added cut is a maximization problem, the inequality $S^{\omega} \leq \sum_{k \geq j+1} \alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega}$ becomes equality. Accordingly, the objective function of the master problem is replaced by $\sum_i w_{ij}x_i + \sum_{\omega \in \Omega_j} p_j[\sum_{k \geq j+1} \alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega}]$. If the constants are eliminated, then the objective function can be $\sum_i w_{ij}x_i - \sum_{\omega \in \Omega_j} p_j[\sum_i \beta_i^{\omega} x_i]$. By using the value of the dual variables β_i^{ω}, we develop a stochastic online algorithm with the primal–dual algorithm, which is presented in Section 4.3.

4.3. Stochastic algorithms using primal–dual algorithms

Feldman et al. [6] provided a primal–dual algorithm to obtain a good competitive ratio for the display ads problem. Primal and dual linear programming (LP) formulations for the display ads problem are as follows:
Primal LP

$$\max \sum_{i=1}^{|A|} \sum_{j=1}^{|T|} w_{ij} x_{ij} \tag{45}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_{ij} \leq 1 \quad \forall j \in T \tag{46}$$

$$\sum_{j=1}^{|T|} x_{ij} \leq C_i \quad \forall i \in A \tag{47}$$

$$x_{ij} \geq 0 \quad \forall i \in A, j \in T \tag{48}$$

Dual LP

$$\min \sum_{j=1}^{|T|} \alpha_j + \sum_{i=1}^{|A|} \beta_i \tag{49}$$

$$\text{s.t. } \alpha_j + \beta_i \geq w_{ij} \quad \forall i \in A, j \in T \tag{50}$$

$$\alpha_j \geq 0, \ \beta_j \geq 0 \ \forall i \in A, j \in T \tag{51}$$

The algorithm uses the dual variables β_i to display an advertisement on a slot. First, the dual variables β_i are all initialized to 0. When a slot $j \in T$ arrives online, select an advertisement i that maximizes $w_{ij} - \beta_i$ among the advertisements available, and is displayed on slot j. If $w_{ij} - \beta_i < 0$, then leave slot j unassigned because the solution is infeasible for the dual. If the advertisement is displayed, then set $x_{ij} := 1$, $\alpha_j := w_{ij} - \beta_i$, and β_i is updated with one of the update rules (i.e. greedy, uniform weighting, or exponential weighting). The rules were proposed by Feldman et al. [6] as a means to obtain good competitive ratios. The algorithm proceeds until all slots of T arrive. At each iteration, the primal solution gives a feasible integer solution and the dual solution is also feasible. The value of β_i plays a role in adjusting the weight w_{ij} by increasing β_i as the number of advertisement i displayed increases. Therefore, the managers need to decide an appropriate value for β_i and use it in the algorithm.

This paper presents a stochastic online algorithm that is based on the primal–dual algorithm. The algorithm updates β_i with $\sum_{\omega \in \Omega_j} p^{\omega} \cdot \beta_i^{\omega}$ (known IID model) or $\sum_{\omega \in \Omega_j} p_j \cdot \beta_i^{\omega}$ (random permutation model) that is obtained by the stochastic programming formulation. Compared with β_i, these values can be more appropriate because they reflect stochastic information. A primal–dual algorithm with stochastic information is shown as follows:

Algorithm 4: Primal–dual algorithm with stochastic information

$x_{ij} \leftarrow 0 \ \ \forall i \in A, j \in T$;
$\beta_i \leftarrow 0, \ C_i^{left} \leftarrow C_i \ \ \forall i \in A$;
$t \leftarrow 1$;
while *a new node $j \in T$ arrives* **do**
  Select $i \in A$ which maximizes the value $w_{ij} - \beta_i$ and satisfies $C_i^{left} > 0$;
  if *$i \in A$ is selected* **then**
    $x_{ij} \leftarrow 1$ and $C_i^{left} \leftarrow C_i^{left} - 1$;
    Update β_i by one rule (e.g. greedy, uniform weighting, or exponential weighting);
  end
  if $t \equiv 0 \ (mod\ \Delta)$ **then**
    Solve the stochastic programming formulation at the time of t^{th} slot to obtain β_i^{ω};
    Update either $\beta_i \leftarrow \lambda \cdot \beta_i + (1-\lambda) \cdot \sum_{\omega \in \Omega_j} p^{\omega} \cdot \beta_i^{\omega}$ (known IID) $\forall i \in A$ or $\beta_i \leftarrow \lambda \cdot \beta_i + (1-\lambda) \cdot \sum_{\omega \in \Omega_j} p_j \cdot \beta_i^{\omega}$ (random permutation) $\forall i \in A$ ($0 \leq \lambda \leq 1$);
  end
  $t \leftarrow t+1$;
end

This algorithm selects advertisement $i \in A$ by using updated for values β_i for each slot. If the stochastic programming formulation is solved at the time of each slot, then the algorithm takes much computation time. To shorten the computation time, the algorithm uses the stochastic technique only for every Δ slots. A parameter λ ($0 \leq \lambda < 1$) is introduced to adjust the effect of the stochastic technique. The effect is greater when the value of λ is small. Section 5 provides the numerical results obtained with the stochastic online algorithm.

5. Computational experiments

In this section, the performances of the primal–dual algorithm and primal–dual algorithm with stochastic information were analyzed. The algorithms were run with JAVA language in Windows 7 on a PC with an Intel® Core™ i5-4690 CPU 3.5 GHz with 16.00 GB of RAM. IBM ILOG CPLEX version 12.8 was used to obtain the dual variables β_i^{ω} for each scenario. For this experiment, we use the 'experimental ratio' as the ratio of the value obtained by Algorithm 4 to the optimal offline objective value at each bipartite graph. We used the following instances in this experiment: 50 advertisements ($|A| = 50$) and 200 slots ($|T| = 200$). The capacity of each advertisement was set to an integer between 1 and 4. The

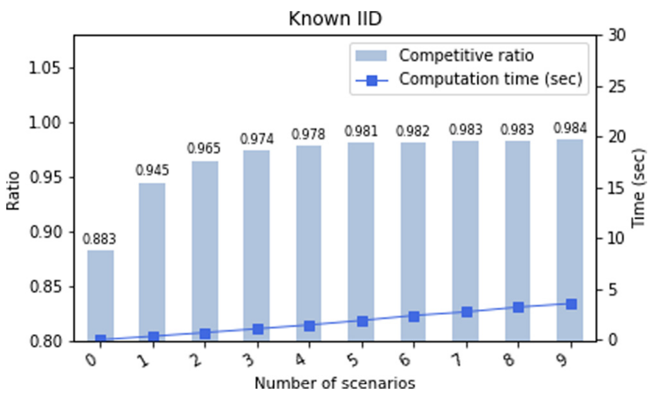

Fig. 1. Results for different value of $|\Omega_j|$ (Known IID).

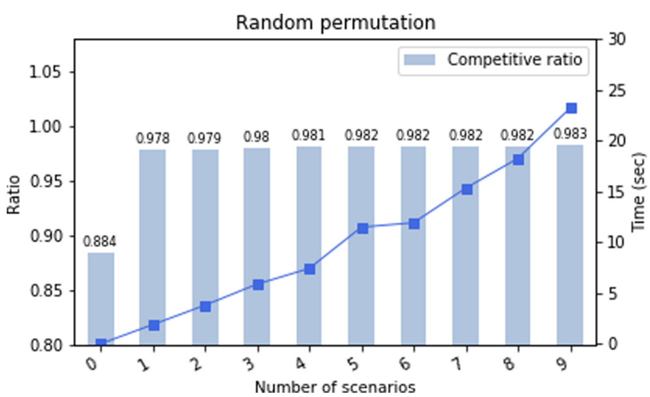

Fig. 2. Results for different value of $|\Omega_j|$ (Random permutation).

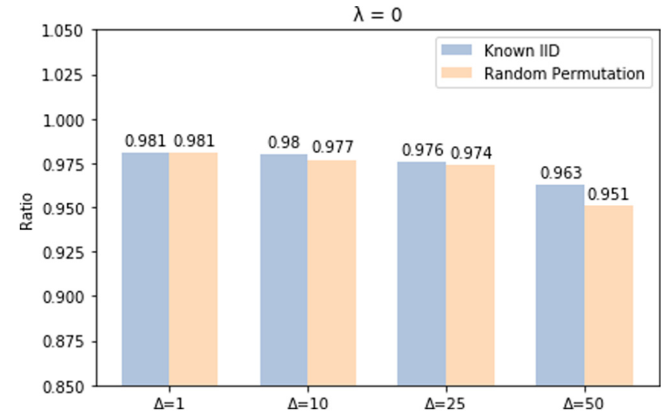

Fig. 3. Experimental ratios for different values of Δ ($\lambda = 0$).

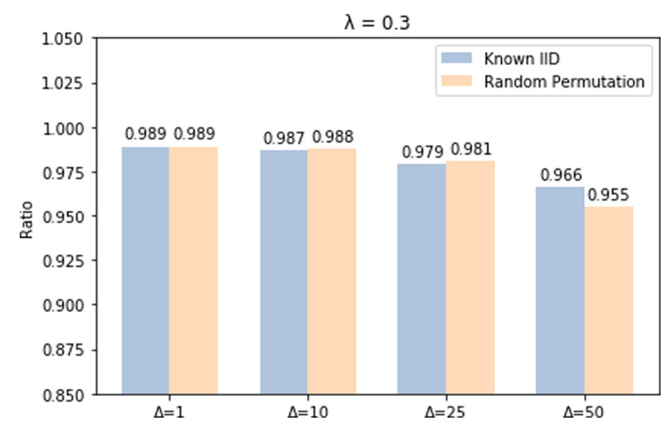

Fig. 4. Experimental ratios for different values of Δ ($\lambda = 0.3$).

ratio of $\frac{U_i}{L_i}$ was limited to 10 for each advertisement. The number of node types was set to 20 ($|K| = 20$) and the sequence of slots follows a multinomial distribution in the known IID model.

To obtain values for β_i^ω, we solve the (dual) slave problem for each scenario. Many future scenarios are drawn for the time of each slot. If we consider all the scenarios drawn, the computation time might be time-consuming for the (dual) slave problems. Figs. 1 and 2 present the experimental ratios and computation times for different values of $|\Omega_j|$. We used $\lambda = 0$ and $\Delta = 1$. The values in Figs. 1 and 2 mean the average values for 100 data sets.

The experimental ratio increased by 6.2% (known IID) and 9.4% (random permutation) when we apply Algorithm 4 ($|\Omega_j| = 1$). The computation times increased linearly in both of the two probabilistic models as $|\Omega_j|$ increases, but the increasing rate was higher in the random permutation model. The experimental ratio tended to increase with increasing values of $|\Omega_j|$. The increase in the experimental ratio was shown to be more apparent in the known IID model than in the random permutation model. The known IID model showed statistically significant differences between the average experimental ratios from $|\Omega_j| = 0$ to 5. Meanwhile, the random permutation showed statistically significant differences from $|\Omega_j| = 0$ to 4. This implies that the adjusted dual variable β_i^ω calculated by using future scenarios showed better performance than β_i obtained by the previous update rule. However, it does not always mean that the performance improves as the number of future scenarios increases. Hence, we decided to set $|\Omega_j|$ to 5 (the known IID) and 4 (the random permutation) for the following analysis, which created a trade-off between the experimental ratios and the computation times, as presented in Figs. 1 and 2. Sensitivity analysis of the parameters λ and Δ was performed for the two stochastic models to derive meaningful insights about using the algorithm with stochastic information.

Figs. 3–6 present the experimental ratios for different values of the parameters λ and Δ. The values in Figs. 3–6 are average values for 100 data sets. Figs. 3–6 show that the primal–dual algorithm with stochastic information performed better as the value of Δ decreased for the two probabilistic orders. It means that the more frequently we use the adjusted dual variable β_i^ω, the more likely we obtain high experimental ratios. The differences in the experimental ratio were as much as 6.2%, depending on Δ (1 to 50). In addition, the figures show that the experimental ratios were affected by the value of λ. The effect of the value of λ tended to be no higher than that of the value of Δ. Overall, $\lambda = 0.3$ showed the best results for these experiments.

Figs. 7–10 present the computation times for different values of the parameters λ and Δ. The average computation times ranged between 0.03 and 1.95 s (known IID) and between 0.11 and 7.52 s (random permutation). The small value of Δ means that the algorithm takes much computation time because the number of stochastic programming formulations to be solved increases. It is crucial to determine the value of Δ by considering the experimental ratios and computation times. we decided to set λ to 0.3 and Δ to 10 for the following analysis, for the following analysis, which created a trade-off between the experimental ratios and computation times, as presented in Figs. 3–10. The empirical results for different ratios of $\frac{U_i}{L_i}$, $|A|$, and $|T|$ are presented in the next subsections.

5.1. Results for known IID model

Fig. 11 presents the empirical results for different ratios of $\frac{U_i}{L_i}$ (between 10 and 1000). Fig. 11 shows the average values for

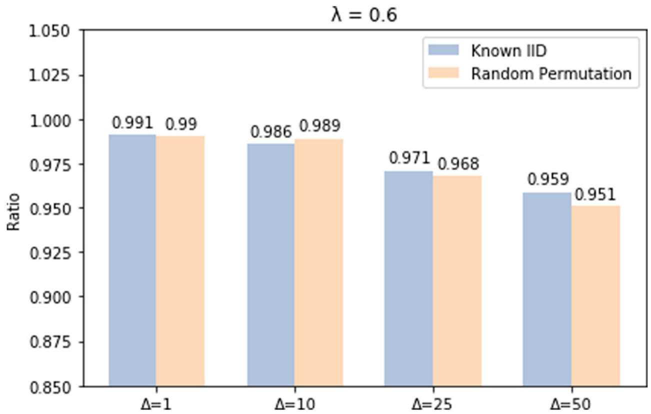

Fig. 5. Experimental ratios for different values of $\varDelta$ ($\lambda = 0.6$).

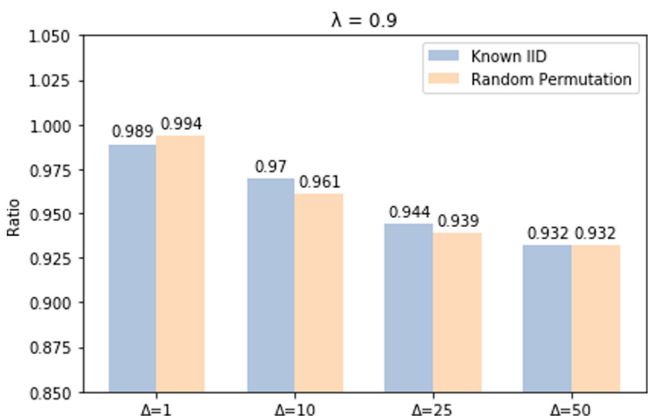

Fig. 6. Experimental ratios for different values of $\varDelta$ ($\lambda = 0.9$).

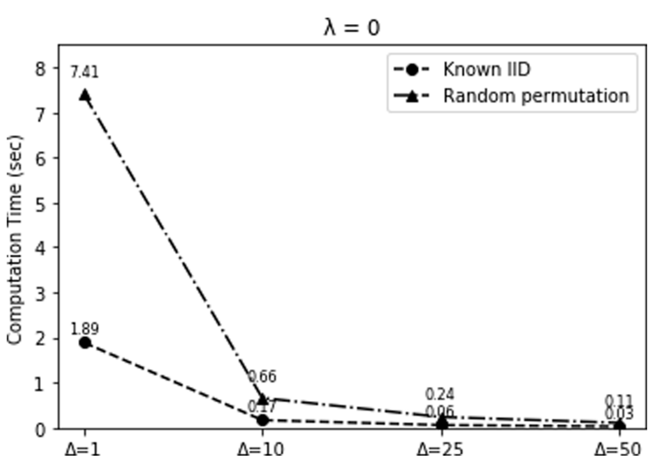

Fig. 7. Computation times for different values of $\varDelta$ ($\lambda = 0$).

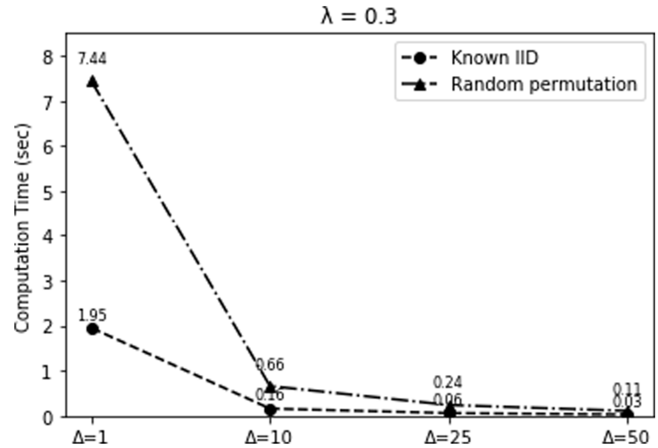

Fig. 8. Computation times for different values of $\varDelta$ ($\lambda = 0.3$).

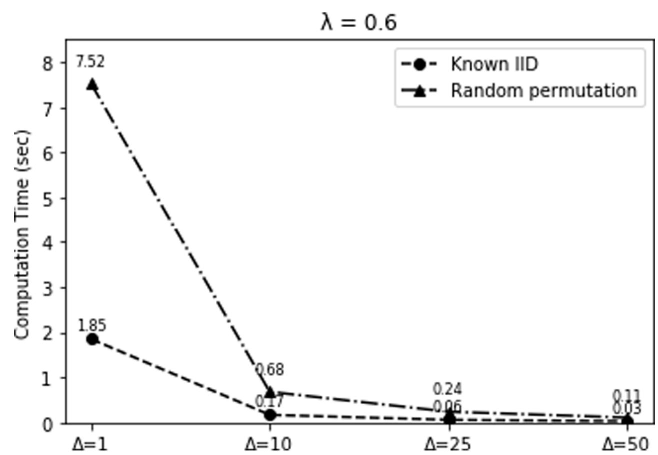

Fig. 9. Computation times for different values of $\varDelta$ ($\lambda = 0.6$).

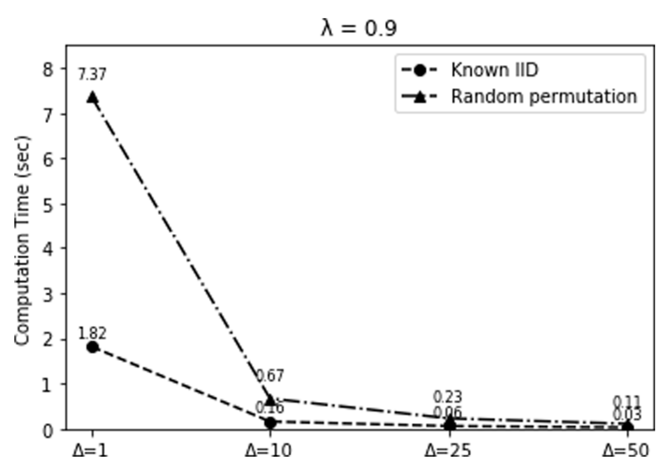

Fig. 10. Computation times for different values of $\varDelta$ ($\lambda = 0.9$).

100 data sets. To compare the performances of the primal–dual algorithm with stochastic information (Algorithm 4), we present the results for the primal–dual algorithm using a greedy update rule as well. The average differences in experimental ratios between the two algorithms ranged between 10.5% and 14.6%. The difference tended to increase as $\frac{U_i}{L_i}$ increased, but the tendency was not high. The ratio of $\frac{U_i}{L_i}$ considerably affects the worst-case bound obtained by Algorithm 3, while it did not highly affect the experimental ratios obtained by Algorithm 4 in the known IID model. Regardless of the ratios of $\frac{U_i}{L_i}$ (between 10 and 1000), the experimental ratios of the primal–dual algorithm with stochastic information showed more than 98%.

Fig. 12 presents the experimental ratios for different values of $|A|$ and $|T|$ (6 cases). Fig. 13 shows the computation times for different values of $|A|$ and $|T|$. The values are average values for 100 data sets under $\frac{U_i}{L_i} = 100$. The experimental ratios tended to slightly increase as $|A|$ and $|T|$ increased in both of the two approaches. This implies that as $|A|$ and $|T|$ increase, advertisements

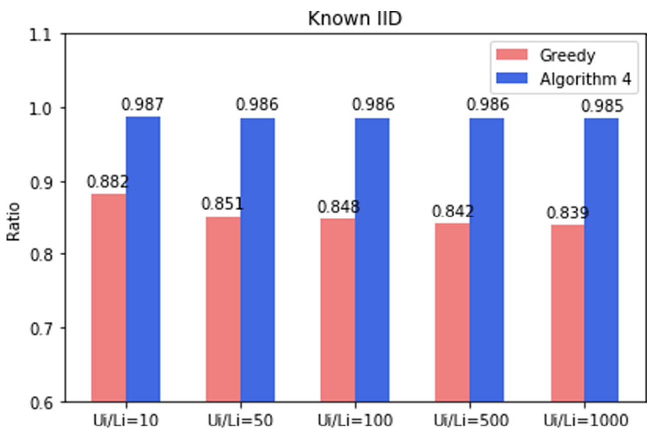

Fig. 11. Results for different ratios of U_i/L_i (Known IID).

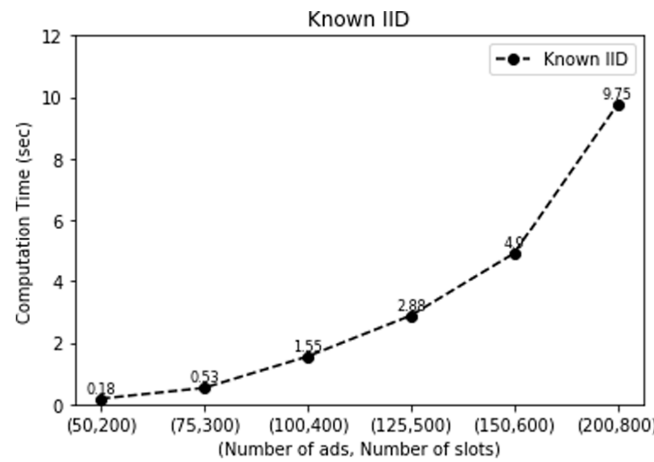

Fig. 13. Computation times for different values of $|A|$ and $|T|$ (Known IID).

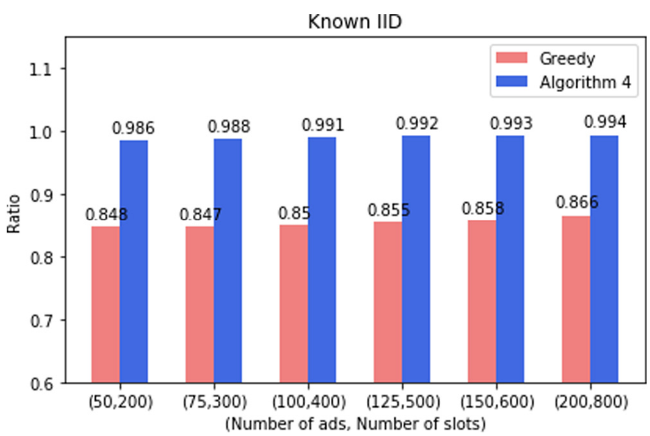

Fig. 12. Experimental ratios for different values of $|A|$ and $|T|$ (Known IID).

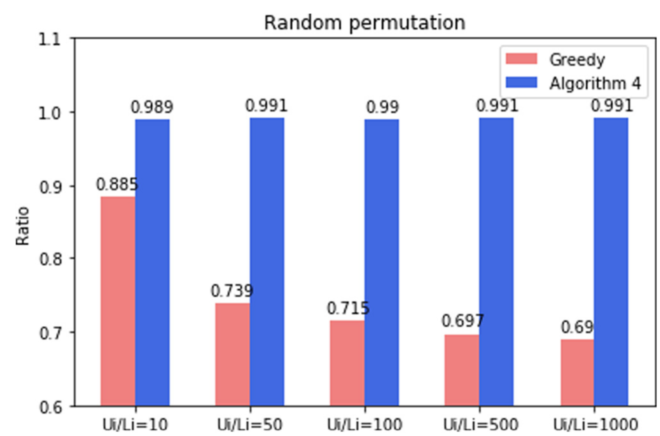

Fig. 14. Results for different ratios of U_i/L_i (Random permutation).

not assigned yet are more likely to have the opportunity to be displayed on the remaining slots. The results from Algorithm 4 showed nearly 99% experimental ratio and the difference of 13 $\sim$ 14% compared to the greedy approach in these experiments. The computation times ranged between 0.18 and 9.75 s and tended to show the tendency of increasing exponentially.

5.2. Results for random permutation model

Fig. 14 presents the empirical results for different ratios of $\frac{U_i}{L_i}$ (between 10 and 1000). Fig. 14 shows the average values for 100 data sets. The average differences in experimental ratios between the two algorithms ranged between 10.5% and 30.1%. The difference tended to increase as $\frac{U_i}{L_i}$ increased. It showed more distinct differences than in the known IID model. Like that for the known IID model, the ratio of $\frac{U_i}{L_i}$ did not highly affect the experimental ratios obtained by Algorithm 4 in the random permutation model. Regardless of the ratios of $\frac{U_i}{L_i}$ (between 10 and 1000), the experimental ratios of Algorithm 4 showed nearly 99%.

Fig. 15 presents the experimental ratios for different values of $|A|$ and $|T|$ (6 cases). Fig. 16 shows the computation times for different values of $|A|$ and $|T|$. The values are average values for 100 data sets under $\frac{U_i}{L_i} = 100$. The experimental ratios tended to slightly increase as $|A|$ and $|T|$ increased in both of the two approaches. Like that for the known IID model, this implies that as $|A|$ and $|T|$ increase, advertisements that had missed the chance of being displayed on the past slots are more likely to have the opportunity to be displayed on the remaining slots. Algorithm 4 showed more than 99% experimental ratios and the difference of 19 $\sim$ 27% compared to the greedy approach in these experiments. The computation times ranged between 0.67 and 102.04 s and tended to show the tendency of increasing exponentially. In these experiments, the random permutation model showed more effective results in terms of experimental ratios, but less efficient results in terms of computation times compared to the known IID model.

5.3. Discussions

This subsection presents managerial insights and limitations for the findings of Algorithms 3 and 4. First, we presented Algorithm 3 to solve the display ads problem without free disposal in the adversarial order. Through mathematical proofs, we showed that Algorithm 3 might be a near-optimal algorithm, according to the value of C_i and $[L_i, U_i]$. The results not only show the improved worst-case guarantees but also lend a theoretical contribution to research on the edge-weighted and capacitated online bipartite matching problem. Notably, the rule for matching between an

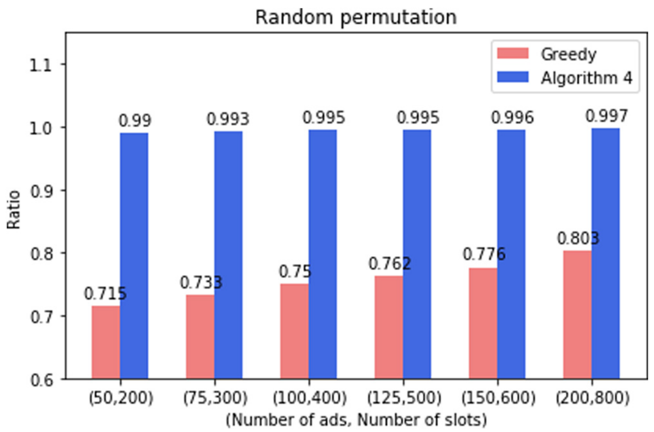

Fig. 15. Experimental ratios for different values of $|A|$ and $|T|$ (Random permutation).

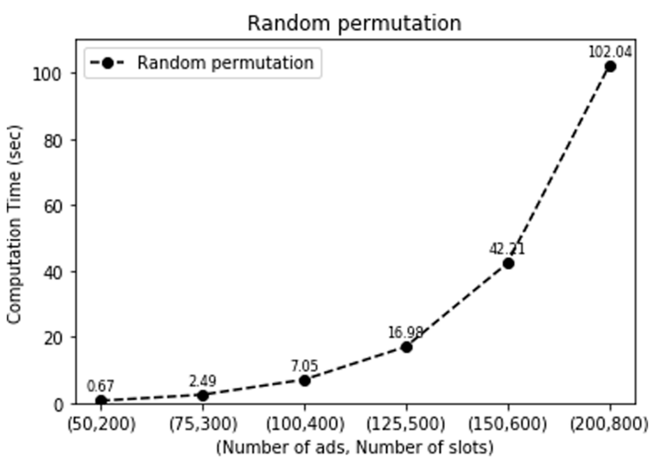

Fig. 16. Computation times for different values of $|A|$ and $|T|$ (Random permutation).

ad and a slot in Algorithm 3 can apply to other types of online bipartite matching problems.

There are some limitations of Algorithm 3. We need weight ranges for advertisements to obtain non-trivial competitive ratios because we excluded the free disposal assumption. The weight ranges might be challenging to know in advance. Also, Algorithm 3 is not a randomized algorithm. We should consider a randomized algorithm with the matching rule in Algorithm 3 to obtain better worst-case guarantees for future research.

Second, Algorithm 4, used in the probabilistic orders, offers some managerial insights. Even though we dealt with the display ads problem considering a strict capacity constraint, the simulation results showed that Algorithm 4 can be a robust approach to solving the problem if we know probabilistic information of the coming slots. Overall, Algorithm 4 found more than 95% ratios (close to 99% ratios in random permutation) in all cases of the numerical experiments. Algorithm 4 was not presented to guarantee worst-case bounds; however, through the simulation results we showed that the algorithm tends to find near-optimal solutions.

There are a lot of future scenarios in each slot. We arbitrarily selected sample scenarios, ($|\Omega_j|$), and solved the stochastic programming problem with those sample scenarios, because it would take a tremendous amount of time to consider all future scenarios. Through the simulation results, we showed more than 97% ratios, even in a small number of scenarios (4-5). There is a possibility that a small number of scenarios lead to bias. However, the simulation results found that the bias might not have a significant impact on the experimental ratios.

It is crucial to decide appropriate values of Δ and λ when we use Algorithm 4 to solve the problem. The parameter Δ is related to the trade-off between effectiveness and efficiency. The parameter λ is used to adjust the effect of the stochastic technique. We found that using only the adjust dual variable β_i^{ω} ($\lambda = 0$) does not always lead to better results through the experiments. Thus, it is recommended that managers decide appropriate values of Δ and λ through their own simulation results.

From a practical point of view, the simulation results from Algorithm 4 can apply to a wide range of fields. For example, the managers would need Algorithm 4 to solve the online advertising assignment problem in which some advertisers are sensitive to the number of times their ads are displayed. It would also be helpful to solve other types of problems (e.g., scheduling or resource allocation) that can correspond to the online bipartite matching problem [35,42]. In these problems, the number of resources (e.g., humans, machines, hotel rooms, and so on) is limited, which has to present a strict capacity constraint.

There are some limitations of Algorithm 4. The first is that we need to estimate probabilistic information of arriving slots as accurately as possible. We need machine learning techniques to infer the probability distribution of arriving slots. Also, when we use Algorithm 4, the results might be different depending on $|\Omega_j|$, Δ, and λ. It might be a difficult task to find appropriate values without knowing historical data and conducting simulations.

6. Conclusions

This paper describes the display ads problem, which is a generalization of the edge-weighted and capacitated online bipartite matching problem. Unlike the existing literature, this paper presents the problem without a free disposal assumption to avoid undermining fairness between advertisements. To obtain bounded competitive ratios, we assume that the online algorithms know the range of the weights $[L_i, U_i]$ for each advertisement i. This paper presents the analyses of the two online input orders (the adversarial and probabilistic orders) to the problem. For the adversarial order, the deterministic online algorithms with worst-case guarantees are proposed. For the probabilistic order, the stochastic online algorithm with scenario-based stochastic programming and Benders decomposition is proposed to solve the problem.

The proposed solution methodologies this paper presents showed good performances in solving the problem. The deterministic online algorithm showed good performances for the analysis with the upper bound on the competitive ratio. The online stochastic algorithms provided better performances than the primal–dual algorithm did through the numerical experiments of the two probabilistic orders. The appropriate values of λ and Δ must be chosen according to the trade-off between the competitive ratio and computation time. The solution methodologies provided good and realistic solutions in the real-time environment of the assignment problem. Therefore, we expect that the algorithms would be useful for managers who work on assigning online advertisements for a website.

CRediT authorship contribution statement

Gwang Kim: Conceptualization, Data curation, Investigation, Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Ilkyeong Moon:** Conceptualization, Supervision, Validation, Writing - review & editing.

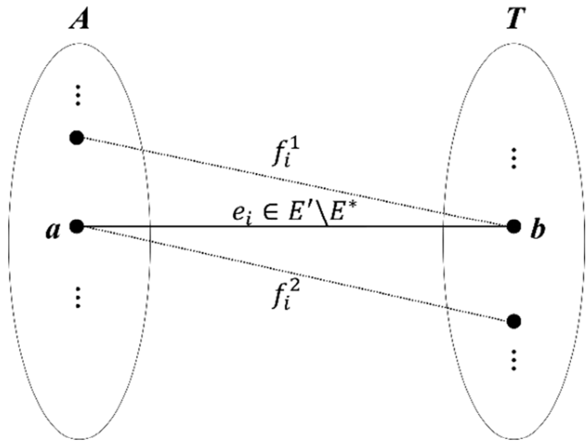

Fig. 17. Example edge in $\mathbf{E}'\backslash\mathbf{E}^*$.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful for the valuable comments from the guest editor and anonymous reviewers. This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning [Grant number NRF-2019R1A2C2084616].

Appendix. Proof of theorems

A.1. Proof of Theorem 1

For simplicity, we assume that all capacities in A have the value of 1. Let $\mathbf{E}^*$ denote the set of optimal edges given by the offline algorithm, and let $\mathbf{E}'$ denote the set of edges produced by Algorithm 1. Let OPT and ALG be the objective values obtained by the offline and Algorithm 1, respectively.

The edges in $\mathbf{E}'$ can be divided into two types: $\mathbf{E}'\cap\mathbf{E}^*$ and $\mathbf{E}'\backslash\mathbf{E}^*$. The total value of the weights from $\mathbf{E}'\cap\mathbf{E}^*$ is the same in OPT and in ALG (value is K). For every edge $e_i \in \mathbf{E}'\backslash\mathbf{E}^*$, there may exist at most two edges f_i^1 and f_i^2 that are for $\mathbf{E}^*\backslash\mathbf{E}'$ as shown in Fig. 17. ($f_i^1(f_i^2)$: the edges incident with $b(a)$ in $\mathbf{E}^*\backslash\mathbf{E}'$, respectively). Let ALG_i denote the weight of the edge e_i and OPT_i denote the total weight obtained from the edges f_i^1 and f_i^2. We obtain $ALG_i \geq L_i$ and $OPT_i < ALG_i + U_i$. It follows that $\frac{ALG_i}{OPT_i} > \frac{L_i}{L_i+U_i}$.

Now we analyze $\mathbf{E}^*\backslash\mathbf{E}'$. Let p denote $|\mathbf{E}'\backslash\mathbf{E}^*|$, noting that $\mathbf{E}^*\backslash\mathbf{E}' \subset \bigcup_{i=1,\ldots,p}(f_i^1 \cup f_i^2)$. Otherwise, the two nodes adjacent to the edge e ($e \in \mathbf{E}^*\backslash\mathbf{E}'$ but $e \notin \bigcup_{i=1,\ldots,p}(f_i^1 \cup f_i^2)$) have no degree in $\mathbf{E}'$. This finding contradicts the procedure used for Algorithm 1 because, in this case, the edge e would be included in $\mathbf{E}'$ while Algorithm 1 proceeds. Therefore, $ALG \geq K + \sum_{i=1,\ldots,p} L_i$ and $OPT < K + \sum_{i=1,\ldots,p}(L_i + U_i)$. According to these inequalities, we obtain

$$\frac{ALG}{OPT} > \frac{K + \sum_{i=1,\ldots,p} L_i}{K + \sum_{i=1,\ldots,p}(L_i + U_i)} \geq \frac{\sum_{i=1,\ldots,p} L_i}{\sum_{i=1,\ldots,p}(L_i + U_i)}$$
$$\geq \min\left(\frac{L_i}{L_i + U_i}\right) = \frac{1}{1 + \max(U_i/L_i)} = \frac{1}{1 + M_1}$$

A.2. Proof of Theorem 2

Let $\mathbf{E}^*$ denote the set of optimal edges given by the offline algorithm and $\mathbf{E}'$ denote the set of edges produced by Algorithm 2. Let OPT and ALG be the objective values obtained by the offline and Algorithm 2, respectively. For Algorithm 2, each $i \in A$ is decomposed into k_i nodes (we call them sub_ads). Each sub_ad has one of the k_i disjoint ranges within $[L_i, U_i]$ and is matched to at most $\lfloor \frac{C_i}{k_i} \rfloor$ nodes in T. Let $\mathbf{E}''$ denote the set of edges produced by Algorithm 2 using the sub_ads.

We note that each edge in $\mathbf{E}^*$ can be mapped to one node (or sub_ad) matched in $\mathbf{E}''$. Let $e_{ij} \in \mathbf{E}^*$ and $w_{ij} \in \left[L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}, L_i(\frac{U_i}{L_i})^{\frac{p+1}{k_i}}\right)$. There are two cases when $j \in T$ arrives: First, $x_{i,p}$ is less than $\lfloor \frac{C_i}{k_i} \rfloor$. In this case, the algorithm can match j to a sub_ad $s_1 \in A$ corresponding to $x_{u,v}$ (which may be $x_{i,p}$). We map e_{ij} to node $j \in T$. It follows that $w_{ij} \leq w_{s_1 j}$. Second, $x_{i,p}$ is $\lfloor \frac{C_i}{k_i} \rfloor$. We map e_{ij} to sub_ad $s_2 \in A$ corresponding to $x_{i,p}$. Let $\mathbf{E}''(s_2) = \{e_{s_2 j} | e_{s_2 j} \in \mathbf{E}''\}$. In this instance, $\lfloor \frac{C_i}{k_i} \rfloor$ edges in $\mathbf{E}''(s_2)$ have already been matched and each edge has a weight of more than $L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}$. Let $ALG(\mathbf{E}''(s_2))$ be the total weight of the edges in $\mathbf{E}''(s_2)$. It follows that $ALG(\mathbf{E}''(s_2)) \geq \lfloor \frac{C_i}{k_i} \rfloor \cdot L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}$.

The edges in $\mathbf{E}^*$ can be divided into two types: $\mathbf{E}_1^*$ and $\mathbf{E}_2^*$ For $\mathbf{E}_1^*$, the edges are mapped from the first case and $\mathbf{E}_2^* = \mathbf{E}^*\backslash\mathbf{E}_1^*$. Let $OPT(\mathbf{E}_1^*)$ and $OPT(\mathbf{E}_2^*)$ be the total weight of the edges for $\mathbf{E}_1^*$ and $\mathbf{E}_2^*$, respectively. For the first case, we map $e_{ij} \in \mathbf{E}_1^*$ to node $j \in T$ and any two edges in $\mathbf{E}_1^*$ cannot be mapped to the same node in T. We have $w_{ij} \leq w_{s_1 j}$. Therefore, $OPT(\mathbf{E}_1^*) \leq ALG$. In the second case, we have $ALG(\mathbf{E}''(s_2)) \geq \lfloor \frac{C_i}{k_i} \rfloor \cdot L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}$. As $w_{ij} \leq L_i(\frac{U_i}{L_i})^{\frac{p+1}{k_i}}$, $ALG(\mathbf{E}''(s_2)) \geq \lfloor \frac{C_i}{k_i} \rfloor \cdot L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}} \geq \lfloor \frac{C_i}{k_i} \rfloor \cdot (\frac{U_i}{L_i})^{-\frac{1}{k_i}} \cdot w_{ij}$. It follows that $w_{ij} \leq \lfloor \frac{C_i}{k_i} \rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}} \cdot ALG(\mathbf{E}''(s_2))$. Note that each $i \in A$ matches at most C_i nodes in T. We have $\sum_{j\in T|e_{ij}\in\mathbf{E}_2^*} w_{ij} \leq C_i \cdot \lfloor \frac{C_i}{k_i} \rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}} \cdot ALG(\mathbf{E}''(s_2)) = M_2 \cdot ALG(\mathbf{E}''(s_2))$. Because $\sum_{s_2\in A} ALG(\mathbf{E}''(s_2)) = ALG$, $\sum_{i\in A}\sum_{j\in T|e_{ij}\in\mathbf{E}_2^*} w_{ij} = OPT(\mathbf{E}^*(s_2)) \leq M_2 \cdot ALG$. We know $OPT = OPT(\mathbf{E}_1^*) + OPT(\mathbf{E}_2^*)$. Therefore, $OPT \leq (1 + M_2) \cdot ALG$ and the competitive ratio can be $\frac{1}{1+M_2}$.

A.3. Proof of Theorem 4

Theorem 3 proves to be $\frac{1}{2}$. First, we define a deterministic algorithm DA. Let OPT and ALG be the objective values obtained by the offline and deterministic algorithm DA, respectively. Next, we prove K_1 using an instance. Assume that $A = \{i\}$ and its capacity is C_i. A sequence $(j_1, j_2, \ldots, j_{(C_i+1)})$ of $C_i + 1$ nodes in T arrives online advertisement i and all nodes in T are adjacent. Let $w_{ij_k} = L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}$ for $1 \leq k \leq C_i + 1$. For any deterministic algorithm, we could find a p value such that j_p is not matched to i. When j_p is not matched to i in DA, the adversary stops the input sequence. Consider an instance with $T = \{j_1, j_2, \ldots, j_p\}$ arriving online. If $p = 1$, $ALG = 0$. It follows that the competitive ratio of this instance would be 0. If $2 \leq p \leq C_i$, $ALG = \sum_{k=1}^{p-1} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}$ and $OPT = \sum_{k=1}^{p} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}$. The ratio is

$$\frac{ALG}{OPT} = \frac{\sum_{k=1}^{p-1} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}}{\sum_{k=1}^{p} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}} \leq \frac{\sum_{k=1}^{p-1} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}}{\sum_{k=2}^{p} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}} = \frac{1}{(\frac{U_i}{L_i})^{\frac{1}{C_i}}} = K_1.$$

If $p = C_i + 1$, then

$$\frac{ALG}{OPT} = \frac{\sum_{k=1}^{p-1} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}}{\sum_{k=2}^{p} L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}} = \frac{1}{(\frac{U_i}{L_i})^{\frac{1}{C_i}}} = K_1.$$

Therefore, an upper bound of the competitive ratio can be K_1.

We also show the way to derive K_2. Assume that $A = \{i\}$ with capacity C_i. There are $\lceil \ln \frac{U_i}{L_i} \rceil$ types of nodes in T ($L_i \neq U_i$). Each type has C_i nodes that arrives continuously. A sequence $(j_1, j_1, \ldots, j_1, j_2, j_2, \ldots, j_2, \ldots, j_{\lceil \ln \frac{U_i}{L_i} \rceil}, j_{\lceil \ln \frac{U_i}{L_i} \rceil}, \ldots, j_{\lceil \ln \frac{U_i}{L_i} \rceil})$ of nodes in T arrives online and advertisement i and all nodes in T are adjacent. Let $w_{ij_k} = L_i(e)^{k-1}$ for $1 \le k \le \lceil \ln \frac{U_i}{L_i} \rceil$. The adversary for any deterministic algorithm DA proceeds as follows: (Define $Y := \lceil \ln \frac{U_i}{L_i} \rceil$)

(1) C_i nodes corresponding to j_1 arrive online: If the number of nodes matched to i (say x_1) is less than or equal to C_i/Y, the adversary stops the input sequence. In other words, $x_1 \le C_1/Y$. The competitive ratio can be $\frac{ALG}{OPT} \le \frac{L_i \cdot (C_i/Y)}{L_i \cdot C_i} = \frac{1}{Y} \le K_2$. Otherwise, the adversary continues in (2).

(2) C_i nodes corresponding to j_2 arrive online: If the number of nodes matched to i (say $x_1 + x_2$) is less than or equal to $2 \cdot C_i/Y$, the adversary stops the input sequence. Because $x_1 > C_i/Y$ and $x_1 + x_2 \le 2 \cdot C_i/Y$, we have $x_2 \le C_i/Y$. Then ALG is at most $L_i \cdot (C_i/Y) + L_i \cdot e \cdot (C_i/Y)$. The competitive ratio can be $\frac{ALG}{OPT} \le \frac{L_i \cdot (e+1) \cdot (C_i/Y)}{L_i \cdot e \cdot C_i} = \frac{1}{Y} \cdot \frac{e+1}{e} \le K_2$. Otherwise, the adversary continues in (s), where s = 3.

(s) ($3 \le s \le Y-1$) C_i nodes corresponding to j_s arrive online: If the number of nodes matched to i (say $\sum_{l=1}^{s} x_l$) is less than or equal to $s \cdot C_i/Y$, the adversary stops the input sequence. As $x_1 > C_i/Y$, $x_1 + x_2 > 2 \cdot C_i/Y$, ..., $x_1 + x_2 + \cdots + x_{s-1} > (s-1) \cdot C_i/Y$, and $x_1 + x_2 + \cdots + x_s \le (s) \cdot C_i/Y$, we have $x_s \le C_i/Y$. Then ALG is at most $L_i \cdot (C_i/Y) + L_i \cdot e \cdot (C_i/Y) + \cdots + L_i \cdot e^{s-1} \cdot (C_i/Y)$. The competitive ratio can be $\frac{ALG}{OPT} \le \frac{L_i \cdot (e^{s-1} + \cdots + e + 1) \cdot (C_i/Y)}{L_i \cdot e^{s-1} \cdot C_i} = \frac{1}{Y} \cdot \frac{e^{s-1} + \cdots + e + 1}{e^{s-1}} \le K_2$. Otherwise, the adversary continues in step (s + 1).

(Y) C_i nodes corresponding to j_Y arrive online: By definition, $x_1 + x_2 + \cdots + x_Y \le C_i = Y \cdot (C_i/Y)$. The competitive ratio can be $\frac{ALG}{OPT} \le \frac{L_i \cdot (e^{Y-1} + \cdots + e + 1) \cdot (C_i/Y)}{L_i \cdot e^{Y-1} \cdot C_i} = \frac{1}{Y} \cdot \frac{e^{Y-1} + \cdots + e + 1}{e^{Y-1}} \le K_2$. Therefore, an upper bound of the competitive ratio can be K_2. (*$\lim_{n \to \infty} 1 + \frac{1}{e} + \cdots + \frac{1}{e^n} = \frac{e}{e-1}$).

References

[1] IAB, IAB Internet Advertising Revenue Report 2018 Full-Year Results, Technical Report. Market Research Report, Interactive Advertising Bureau, 2018.

[2] P. Manik, A. Gupta, P.C. Jha, K. Govindan, A goal programming model for selection and scheduling of advertisements on online news media, Asia-Pac. J. Oper. Res. 33 (02) (2016) 1650012.

[3] R. Gopal, X. Li, R. Sankaranarayanan, Online keyword based advertising: Impact of ad impressions on own-channel and cross-channel click-through rates, Decis. Support Syst. 52 (1) (2011) 1–8.

[4] A. Mehta, Online matching and ad allocation, Found. Trends® Theoret. Comput. Sci. 8 (4) (2013) 265–368.

[5] S. Bhat, M. Bevans, S. Sengupta, Measuring users' web activity to evaluate and enhance advertising effectiveness, J. Advert. 31 (3) (2002) 97–106.

[6] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, M. Pál, Online ad assignment with free disposal, in: International Workshop on Internet and Network Economics, Springer, 2009, pp. 374–385.

[7] A. Khatibi, G. Baharian, E.R. Kone, S.H. Jacobson, The sequential stochastic assignment problem with random success rates, IIE Trans. 46 (11) (2014) 1169–1180.

[8] S.A. Erdogan, A. Gose, B.T. Denton, Online appointment sequencing and scheduling, IIE Trans. 47 (11) (2015) 1267–1286.

[9] X. Qiu, H.Y.K. Lau, An AIS-based hybrid algorithm with PDRs for multi-objective dynamic online job shop scheduling problem, Appl. Soft Comput. 13 (3) (2013) 1340–1351.

[10] C.C. Aggarwal, J.L. Wolf, S.Y. Philip, A framework for the optimizing of WWW advertising, in: Trends in Distributed Systems for Electronic Commerce, Springer, 1998, pp. 1–10.

[11] B. Birnbaum, C. Mathieu, On-line bipartite matching made simple, Acm Sigact News 39 (1) (2008) 80–87.

[12] M. Chang, D.S. Hochbaum, Q. Spaen, M. Velednitsky, An optimally-competitive algorithm for maximum online perfect bipartite matching with iid arrivals, Theory Comput. Syst. (2019) 1–17.

[13] X.A. Chen, Z. Wang, A dynamic learning algorithm for online matching problems with concave returns, European J. Oper. Res. 247 (2) (2015) 379–388.

[14] D.F. Ciocan, V.F. Farias, Dynamic allocation problems with volatile demand, Math. Oper. Res. 37 (3) (2012) 501–525.

[15] J. Feldman, A. Mehta, V. Mirrokni, S. Muthukrishnan, Online stochastic matching: Beating 1-1/e, in: 2009 50th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2009, pp. 117–126.

[16] G. Goel, A. Mehta, Online budgeted matching in random input models with applications to adwords, in: SODA, vol. 8, 2008, pp. 982–991.

[17] B. Haeupler, V.S. Mirrokni, M. Zadimoghaddam, Online stochastic weighted matching: Improved approximation algorithms, in: International Workshop on Internet and Network Economics, Springer, 2011, pp. 170–181.

[18] Z. Huang, Z.G. Tang, X. Wu, Y. Zhang, Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals, 2018, arXiv preprint arXiv:1804.07458.

[19] P. Jaillet, X. Lu, Online stochastic matching: New algorithms with better bounds, Math. Oper. Res. 39 (3) (2014) 624–646.

[20] K. Kang, K. Altinkemer, I. Hur, Mobile coupons delivery problem: Postponable online multi-constraint knapsack, Decis. Support Syst. 116 (2019) 13–25.

[21] C. Karande, A. Mehta, P. Tripathi, Online bipartite matching with unknown distributions, in: Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, 2011, pp. 587–596.

[22] R.M. Karp, U.V. Vazirani, V.V. Vazirani, An optimal algorithm for on-line bipartite matching, in: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, 1990, pp. 352–358.

[23] M. Mahdian, Q. Yan, Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps, in: Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, 2011, pp. 597–606.

[24] V.H. Manshadi, S.O. Gharan, A. Saberi, Online stochastic matching: Online actions based on offline statistics, Math. Oper. Res. 37 (4) (2012) 559–573.

[25] J. Naor, D. Wajc, Near-optimum online ad allocation for targeted advertising, ACM Trans. Econ. Comput. (TEAC) 6 (3–4) (2018) 1–20.

[26] A. Torrico, S. Ahmed, A. Toriello, A polyhedral approach to online bipartite matching, Math. Program. 172 (1–2) (2018) 443–465.

[27] A. Hertz, R. Montagné, F. Gagnon, Online algorithms for the maximum k-colorable subgraph problem, Comput. Oper. Res. 91 (2018) 209–224.

[28] N. Korula, M. Pál, Algorithms for secretary problems on graphs and hypergraphs, in: International Colloquium on Automata, Languages, and Programming, Springer, 2009, pp. 508–520.

[29] T. Kesselheim, K. Radke, A. Tönnis, B. Vöcking, An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions, in: European Symposium on Algorithms, Springer, 2013, pp. 589–600.

[30] J. Feldman, M. Henzinger, N. Korula, V.S. Mirrokni, C. Stein, Online stochastic packing applied to display ad allocation, in: European Symposium on Algorithms, Springer, 2010, pp. 182–194.

[31] A. Bhalgat, J. Feldman, V. Mirrokni, Online allocation of display ads with smooth delivery, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1213–1221.

[32] H.F. Ting, X. Xiang, Near optimal algorithms for online maximum edge-weighted b-matching and two-sided vertex-weighted b-matching, Theoret. Comput. Sci. 607 (2015) 247–256.

[33] X. Sun, J. Zhang, J. Zhang, Near optimal algorithms for online weighted bipartite matching in adversary model, J. Comb. Optim. 34 (3) (2017) 689–705.

[34] S. Agrawal, Z. Wang, Y. Ye, A dynamic near-optimal algorithm for online linear programming, Oper. Res. 62 (4) (2014) 876–890.

[35] U. Bhaskar, A. Jalal, R. Vaze, The adwords problem with strict capacity constraints, in: 36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[36] N. Buchbinder, K. Jain, J.S. Naor, Online primal-dual algorithms for maximizing ad-auctions revenue, in: European Symposium on Algorithms, Springer, 2007, pp. 253–264.

[37] D. Charles, M. Chickering, N.R. Devanur, K. Jain, M. Sanghi, Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads, in: Proceedings of the 11th ACM Conference on Electronic Commerce, 2010, pp. 121–128.

[38] N.R. Devanur, T.P. Hayes, The adwords problem: online keyword matching with budgeted bidders under random permutations, in: Proceedings of the 10th ACM Conference on Electronic Commerce, 2009, pp. 71–78.

[39] N.R. Devanur, K. Jain, B. Sivan, C.A. Wilkens, Near optimal online algorithms and fast approximation algorithms for resource allocation problems, J. ACM 66 (1) (2019) 1–41.

[40] P. Jaillet, X. Lu, Near-optimal online algorithms for dynamic resource allocation problems, 2012, arXiv preprint arXiv:1208.2596.

[41] B. Kalyanasundaram, K.R. Pruhs, An optimal deterministic algorithm for online b-matching, Theoret. Comput. Sci. 233 (1–2) (2000) 319–325.

[42] A. Legrain, P. Jaillet, A stochastic algorithm for online bipartite resource allocation problems, Comput. Oper. Res. 75 (2016) 28–37.

[43] A. Mehta, A. Saberi, U. Vazirani, V. Vazirani, Adwords and generalized online matching, J. ACM 54 (5) (2007) 22–es.

[44] V.S. Mirrokni, S.O. Gharan, M. Zadimoghaddam, Simultaneous approximations for adversarial and stochastic online budgeted allocation, in: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2012, pp. 1690–1701.

[45] J.F. Benders, Partitioning procedures for solving mixed-variables programming problems, Comput. Manag. Sci. 2 (1) (2005) 3–19.