Omega 40 (2012) 562–570

Contents lists available at SciVerse ScienceDirect

Omega

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

Scheduling online advertisements to maximize revenue under variable display frequency

Jason Deane [a,*], Anurag Agarwal [b,1]

[a] Department of Business Information Technology, 1007 Pamplin Hall, College of Business, Virginia Tech, Blacksburg, VA 24061, USA
[b] Department of Information Systems and Decision Sciences, College of Business, University of South Florida, Sarasota, FL 34243, USA

A R T I C L E   I N F O

Article history:
Received 15 April 2009
Accepted 1 November 2011
Processed by Associate Editor Geunes
Available online 18 November 2011

Keywords:
Scheduling
Optimization
Combinatorial Analysis

A B S T R A C T

The online advertising industry realized annual revenues estimated at over $26 billion, in the United States alone, in 2010. Banner advertising accounts for an estimated 23% of all online advertising revenues. Publishers of banner advertisements face a scheduling optimization problem on a daily basis. Several papers in the literature have proposed mathematical models and solution approaches to address a publisher's banner advertisement scheduling problem and the problem has been shown to be NP-hard. In this paper we propose a new model variation for the problem, which incorporates variable display frequencies. We find that the variable-display frequency model provides significantly improved space utilization relative to the fixed-display frequency model and consequently higher revenues for the publishers.

1. Introduction

Online advertising revenues generated in the US were estimated at over $26 billion in 2010, up 15% from 2009 revenues [1]. During the first quarter of 2011 alone, the revenues were estimated at $7.3 billion, an increase of 23% over the first-quarter of 2010 [2]. According to Schneider [3], by 2014, the annual revenues are estimated to grow to approximately $34 billion. Revenues from banner advertisements, hereafter referred to as ads, account for nearly a quarter of this revenue stream [2]. Part of this growth can be explained by the fact that many web companies that, in the past, relied on a subscription-based revenue model are shifting to an ad-supported revenue model or a hybrid ad-supported and subscription-based revenue model. Subscription based models generate revenue solely from customers paying subscription fees to access the website content, whereas ad-supported models do so through the selling of ad space on the website. Many companies have found the latter to be more profitable in the recent years. For example Slate.com, an online magazine, first started its business with a subscription-fee-only revenue model, but eventually switched to an ad-supported revenue model as subscriptions alone were unprofitable [3]. Publishers of banner ads face the problem of scheduling ads for their customers on a daily basis. A well-formulated mathematical scheduling model and a sound solution methodology can make a significant difference in the ability of online ad publishers to maximize revenues. In this paper we address the problem of ad scheduling from an optimization perspective and propose a new model that helps improve the ad-space utilization, which translates to higher revenues for the publisher.

Several mathematical models for scheduling banner ads have been proposed in literature; however, by far the most popular model is called the Maxspace problem. Adler et al. [4] first proposed the MaxSpace problem in which the publisher tries to schedule and display the requested frequency of ads for each advertiser such that the available space is best utilized to maximize revenues. This ad-scheduling problem, like most scheduling problems, has been shown to be NP-hard in nature [5]. Dawande et al. [5] and Kumar et al. [6] discussed the related complexity issues and also proposed genetic algorithm approaches for this problem. In the original MaxSpace formulation, if the requested frequency of ad impressions for a given customer cannot be displayed due to space limitations, then none of the impressions for that customer are displayed. Kumar et al. [6,7] and Freund and Naor [8] also share this all-or-nothing constraint. Amiri and Mennon [9,10] addressed this issue and extended the original MaxSpace problem formulation by proposing multiple display frequencies. While this offers a slight improvement, it still limits space utilization.

The aforementioned models, in the literature, fail to incorporate some real-world considerations such as the incorporation of an acceptable ad frequency range, the standardization of ad sizes and a realistic length of planning horizon (these observations are

* Corresponding author. Tel.: +1 540 231 4535; fax: +1 540 231 3752.
E-mail addresses: jason.deane@vt.edu (J. Deane), agarwala@sar.usf.edu (A. Agarwal).
[1] Tel.: +1 941 359 4522.

0305-0483/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.omega.2011.11.001

based on the needs of top online advertisement companies.) As a result, companies offering variable display frequencies to their customers are not able to utilize these models. In this paper we extend the work of Adler et al. [4], Dawande et al. [5], Kumar et al. [6] and Amiri and Menon [9,10] by modeling variable display frequencies, which may vary between a lower and an upper bound. Our results indicate that this change could mean significantly more revenue. In addition, we make use of the industry accepted standard ad sizes in the development of our test problems [11]. Any publishers that are not currently offering variable frequencies might use this model as a motivation to consider the variable-frequency option.

Publishers commonly provide the advertisers a pre-defined set of ad sizes to choose from. This practice has been ignored in previous works. Prior studies have instead reported results on problems based on random ad sizes. One could certainly argue that standard ad sizes are a special case of the more general random ad sizes and that as long as the model works well for the general case, it should also work well for the special case. However, we believe that there could be some domain-specific nuances that might be missed if we do not follow industry conventions and test the models on standard ad sizes. In addition, extant literature has focused on a very short planning horizon such as one hour. In reality, ads are scheduled half-a-day, a full day or even two days in advance. In preparing our test problems we incorporate these real-world considerations. We compare the space utilization for the traditional fixed-frequency model with our proposed variable-frequency model empirically using both exact methods (using CPLEX®) and also using several heuristics that we propose in this paper. In addition to offering the publisher and advertisers added flexibility, our experimental results indicate that the proposed variable frequency model also offers a significant improvement in the space utilization, which is a publisher's primary objective.

The rest of the paper is organized as follows. In Section 2, we describe the mathematical formulation of the proposed variable-frequency MaxSpace model. In Section 3, we describe the proposed solution techniques. In Section 4, we provide a discussion of the experimental testing process. In the final section, we provide conclusions and propose ideas for future work in this area.

2. Variable frequency MaxSpace model (VF-MaxSpace)

Similar to the original MaxSpace model introduced by Adler et al. [4], the VF-MaxSpace model is defined for a pre-determined planning period (or horizon), which varies between 1 h and 2 days. In practice the planning period is typically divided into slots of 1 min each. Therefore, a 1-h planning period will have 60 slots while a 1-day planning period will have 1440 slots (a 2-day planning period will have 2880 slots). The scheduling process basically assigns ads to these 1-min time slots. Any user who accesses the publisher's website during a particular time slot will see the ads assigned to that slot. Prior research has focused on planning periods of up to 100 slots. In our numerical studies, we experience that a 100-slot scheduling problem can be solved to optimality using CPLEX® in a fraction of a second. We focus on problems of 720, 1440 and 2880 slots, which are routinely faced by online ad publishers and which often cannot be solved to optimality in a time-efficient manner.

Fig. 1 illustrates an example of the ad-scheduling problem. Each vertical rectangle represents a vertical banner for a time slot. In this example, there are 10 slots representing a planning period of 10 min. The height of a slot is S and ads of various sizes can be assigned to each slot. For example, an ad of size s_1 and an ad of size s_2 are assigned to the first slot.

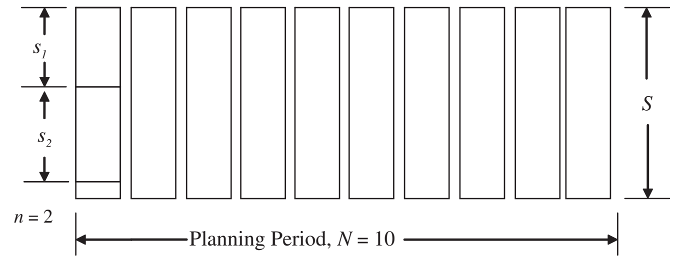


Fig. 1. Illustration of an ad-scheduling problem.

Below, we define the notation used throughout the paper. Additional notation will be defined as needed.

N	number of time slots in the planning period (time slots are indexed by j)
n	number of ads (or customers) for which space has been requested during the planning period (indexed by i)
S	banner height (for vertical banners) or width (for horizontal banners)
s_i	height (for vertical banners) or width (for horizontal banners) of requested ad i, $i=1,2,\ldots,n$
L_i	lower bound on the display frequency of ad i, $i=1,2,\ldots,n$
U_i	upper bound on the display frequency of ad i, $i=1,\ldots,n$
x_{ij}	binary decision variable whose value is 1 if ad i is assigned to slot j, 0 otherwise
y_i	binary variable whose value is 1 if ad i is assigned to at least one time slot, 0 otherwise

2.1. VF-MaxSpace problem

$$Max \sum_{j=1}^{N} \sum_{i=1}^{n} s_i x_{ij} \tag{1}$$

s.t.

$$\sum_{i=1}^{n} s_i x_{ij} \le S, j = 1,2,\ldots,N \tag{2}$$

$$L_i y_i \le \sum_{j=1}^{N} x_{ij} \le U_i y_i, \quad i = 1,2,\ldots,n \tag{3}$$

$$x_{ij} \in \{0,1\}, \quad i = 1,2,\ldots,n;\ j = 1,2,\ldots,N \tag{4}$$

$$y_i \in \{0,1\}, \quad i = 1,2,\ldots,n \tag{5}$$

The objective of the publisher, stated in (1), is to maximize the space used to publish ads, which also maximizes the ad space utilization, assuming a fixed capacity. Since the model assumes linear pricing, maximizing space utilization is equivalent to maximizing revenue for the publisher. The first set of constraints, stated in (2), ensures that the combined height of the set of ads assigned to any slot does not exceed the banner height (width in the case of a horizontal banner). The second set of constraints, stated in (3), enforces the ad display frequency bounds for each ad for which space has been requested. If an ad is selected for assignment, its display frequency must be between the lower bound L_i and the upper bound U_i. Constraints stated in (4) and (5) are the binary constraints for x_{ij} and y_i, respectively.

Display frequency bounds represent a variation from the model solved in Adler et al. [4] and Amiri and Menon [9,10]. Adler et al. [4], Kumar et al. [6,7] and Freund and Naor [8] utilize a bounding constraint stated as $\sum_{j=1}^{N} x_{ij} = w_i y_i, \quad i = 1,2,\ldots,n$. This constraint ensures that the ad, if displayed, is displayed exactly the requested number of times, w_i, over the planning period. In our model we replace this hard all-or-nothing frequency constraint with the variable-frequency constraint, forcing the number of impressions to lie between a lower and an upper bound. The variable-frequency model is motivated by the fact that it helps both the publisher and the advertisers. It improves the space utilization for the publisher and provides a minimal exposure guarantee to the customers. In most cases, having some minimum number of ads displayed is better than none. This allows the advertiser to feel confident that their marketing campaign will achieve at least a certain level of market penetration. It is probably for this reason that many real-world contracts are written in terms of variable frequencies.

The original MaxSpace problem is in some ways similar to the well-known knapsack problem [12–21]. The available ad space can be regarded as the space available in a knapsack within which we are trying to determine which items to place, and the ads with fixed frequencies can be regarded as items to be added in the knapsack. While the two problems have similarities, there are notable differences as well. One of the main differences is the underlying structure of the units being considered and the number of containers included. In the knapsack problem, the items are considered for inclusion as a single unit. In the MaxSpace problem, the ads are not added as a single unit, but instead broken into several smaller units in time slots. This difference is exacerbated in the proposed variable-frequency model because the items (ads) are of variable sizes. The problem also shares many characteristics with the well-researched facility location problem [22–28]. Prior research has shown that the knapsack and the facility location problems can be solved using a variety of techniques such as branch and bound algorithms, fractional cutting plane, lagrangean relaxation and other heuristics [12–28].

The primary focus of this work is to determine the relative power of the proposed variable-frequency model in contrast to its fixed frequency counterpart. In addition to providing better ad-space utilization, the proposed model also affords the ad publishers more flexibility in terms of the types of contracts that may be offered to potential customers. The solution techniques we use to demonstrate the effectiveness of the proposed model are the exact IP approach, using CPLEX® and several single-pass greedy heuristics proposed in this paper. CPLEX® was chosen based on its reputation as one of the most powerful commercially available IP solvers. While CPLEX® is extremely powerful, certain situations or problem instances may necessitate a technique that provides a relatively good solution very fast. For the larger instances of this NP-hard problem, CPLEX® may not be the best option, often running into timing and/or memory issues. In an effort to provide publishers with a few viable alternatives, we developed some single-pass heuristics. These simple heuristics were chosen because they are direct adaptations of well-established heuristics already suggested in the literature for the fixed frequency model [4–6]. We provide computational results for CPLEX and the greedy heuristics for the traditional fixed frequency model and the proposed variable frequency model. Keep in mind that the focus of this work is not to identify the most powerful method by which to solve the problems, but instead to highlight the power of the proposed variable frequency model. This, we are hopeful, will motivate future researchers to develop more efficient and powerful solution techniques. We discuss the various solution approaches in more detail in Section 3.

3. Model solution approaches

For the fixed frequency MaxSpace problem, several single-pass heuristics have been proposed [4–6]. Four of the most commonly used greedy heuristics are the Largest-Volume-Least-Full (LVLF), the Largest-Volume-Most-Full (LVMF), the Smallest-Volume-Least-Full (SVLF) and the Smallest-Volume-Most-Full (SVMF). Of these heuristics, the LVMF has been shown to be the most effective. In the LVMF heuristic, the ad that demands the largest volume (where volume=ad-size × frequency) gets the highest priority and if there are enough slots available for this ad, the ad is assigned to slots that are most full. Since the problems under consideration in this study differ from the original MaxSpace problem in that the display frequencies are variable, building on prior work [4–6], we propose modified versions of the LVMF, LVMF, SVLF and SVMF heuristics, called VF-LVMF, VF-LVMF, VF-SVLF and VF-SVMF, respectively. The modified heuristics have a time complexity of $O(nN\log(N))$. Next we explain the four greedy heuristics.

3.1. Greedy heuristics

Variable Frequency-Largest Volume Least Full (VF-LVLF) Heuristic

1. For each ad i, define $Vol_i = U_i s_i$
2. sort the ads in descending order of Vol_i
3. sort slots in the order of occupied volume, least full to most full
4. for each ad i in the sorted list
   - determine the feasibility of assignment; i.e., check to see if at least L_i number of slots have at least s_i space available;
   - if feasible, assign an impression of ad i to each of the next L_i least-full slots;
   - sort slots in the order of volume occupied, least full to most full;
5. for each ad i in the sorted list
   - assign impressions of ad i to slots until either no space is available or U_i number of impressions are assigned.

Variable Frequency-Largest Volume Most Full (VF-LVMF) Heuristic
Same as VF-LVLF, except that in Steps 3 and 4, sort slots in the order of most full to least full.

Variable Frequency-Smallest Volume Least Full (VF-SVLF) Heuristic
Same as VF-LVLF, except that in Step 2, sort ads in ascending order of Vol_i

Variable Frequency-Smallest Volume Most Full (VF-SVMF) Heuristic
Same as VF-SVLF except that in Steps 3 and 4, Sort slots in the order of most full to least full.

See Appendix A for an example problem. In the example, we show the steps of applying the LVMF and VF-LVMF heuristics for a small, 10-slot, 8-advertiser problem.

3.1.1. Worst-case analysis of the heuristics

In this section, we provide worst case analyses for the heuristics. We first define the worst-case-performance ratio $r = f/f^*$ as the fraction of total available space used in the worst case, where f is the space utilization from the heuristic solution and f^* is the space utilized by the optimal solution. In comparing heuristics, a higher value of r is better. A higher value of r indicates more power in the worst case. Initially, we focus on the fixed frequency heuristics that are based on the original MaxSpace problem, and then describe their relationship to the heuristics for the VF-MaxSpace problem.

3.1.1.1. SVLF and SVMF heuristics. Given our previous notation, N=the number of banners and S=the width of each banner. This results in a total ad space volume of NS. Suppose the number of ads $n=2$ (namely A_1 and A_2). Suppose further that the width of A_1 is the minimum possible ad size $s_1=1$ and that its frequency $w_1=1$, and that ad A_2 has $s_2=S$ and $w_2=N$. Thus, the problem can be solved to optimality by assigning ad A_2 in each of the N slots, giving a space utilization f^* (and a denominator for r) of 100% or 1.

In the heuristic SVLF (or SVMF), A_1 has the least volume and is assigned to the first banner, satisfying its frequency. Because the remaining volume of the banners is $(NS)-1$, and the required volume ($s_i w_i$) for A_2 is NS, no other assignments can be made (remember that in the fixed-frequency heuristics, either all or nothing must be assigned). Thus, in the heuristic solution the space utilization $f=1/(NS)$, $r=f/f^*=1/(NS)$ and as $(NS)\rightarrow\infty$, $r\rightarrow 0$.

3.1.1.2. LVLF and LVMF heuristics. As above, let N=the number of banners and S=the width of each banner such that the total ad space volume is NS. Suppose the number of ads $n=3$ (namely A_1–A_3). Let the width of ads A_1 and A_2 be one-half the banner width and the frequency be the same as the number of banners such that $s_1=s_2=S/2$ and $w_1=w_2=N$. A_3 also has a frequency $w_3=N$ but has a width $s_3=(S/2)+1$. The optimal solution assigns A_1 and A_2 to each banner for a utilization f^* (and a denominator of r) of 100% or 1.

Under heuristic LVLF (or LVMF), A_3 will have the largest volume ($s_i w_i$) and will be assigned to each of the N banners. The remaining width of each banner will be $(S/2)-1$, creating a remaining volume of $N((S/2)-1)$. Because ads A_1 and A_2 both require volume $N(S/2)$, no further assignments will be made. Thus, for the heuristic solution the space utilization $f=((S/2)+1)/S=(1/2)+(1/S)$. Because $r=f/f^*$, $r=(1/2)+(1/S)$. Therefore, as $S\rightarrow\infty$, $r=(1/2)+\varepsilon$ where $\varepsilon\rightarrow 0$.

The worst-case-performance ratios for the variable frequency adaptations of these heuristics for the VF-MaxSpace problem are the same, because the MaxSpace problem is a special case of the VF-MaxSpace problem when $L_i=U_i$.

4. Numerical studies

We first describe the process and the rationale underlying our problem generation process and then present and discuss the results.

4.1. Test problems generation

To test the proposed model and solution approaches, we generated several test problems. Prior work had focused on a limited planning horizon of only up to 100 min [5,6]. A 100-slot problem can be solved to optimality using CPLEX® in a fraction of a second. In this study, we expand the planning horizon to 12 h (720 min), 1 day (1440 min) and 2 day (2880 min). These longer planning periods are routinely encountered in practice. Further, as discussed in Section 1, prior studies have utilized random ad sizes. We chose to use standard ad sizes developed by the Internet Advertising Bureau (IAB) [1]. For example, for horizontal banners, we use the banner width as 800 pixels and the ad sizes as 88×60, 90×60, 105×60, 120×60, 234×60 and 468×60. For vertical banners, we use a total height of 900 pixels, with ad sizes of 120×60, 120×90, 120×150, 120×200, 120×240, 120×250 and 120×280. Note that the IAB specifies many other types of ads such as skyscrapers, leaderboards, micro-bar and rectangles, which are not included in our study. Our focus is limited to horizontal and vertical banner ads.

Also, in prior work, the demand for ad space, i.e., the number of ads multiplied by their requested ad frequencies per planning period, was much higher than the available capacity [6]. As a result, only a fraction of advertisers' requests could be fulfilled. This reduces the problem difficulty and, in reality, such a situation would not persist as economic forces would cause prices and capacity to adjust so that the demand for ad space is somewhat close to the available capacity. The display frequency bounds L_i and U_i were chosen randomly from a uniform distribution with lower and upper limits as shown in Table 1. The number of advertisers, n, and the upper and lower frequency limits were chosen such that the lower limit for each set generated a total demand that is below the available capacity and the upper limit was chosen to ensure that the average demand is higher than the available capacity. Otherwise, the problems would be too easy to solve. The problems thus generated are both realistic and challenging.

Table 1
Description of Problem Sets for the VF problems.

Problem set	Banner type	Planning period (N)	Number of advertisers (n)	Range for L_i and U_i	No. of problems
720H-1	Horiz	720	10–20	100–200	25
720H-2	Horiz	720	20–30	50–150	25
720H-3	Horiz	720	50–80	10–100	25
720V-1	Vert	720	10–20	100–200	25
720V-2	Vert	720	20–30	50–150	25
720V-3	Vert	720	50–80	10–100	25
1440H-1	Horiz	1440	20–30	200–300	25
1440H-2	Horiz	1440	30–50	100–200	25
1440H-3	Horiz	1440	60–100	10–100	25
1440V-1	Vert	1440	20–30	200–300	25
1440V-2	Vert	1440	30–50	100–200	25
1440V-3	Vert	1440	60–100	10–100	25
2880H-1	Horiz	2880	20–30	300–500	25
2880H-2	Horiz	2880	30–50	200–400	25
2880H-3	Horiz	2880	60–100	50–200	25
2880V-1	Vert	2880	20–30	300–500	25
2880V-2	Vert	2880	30–50	200–400	25
2880V-3	Vert	2880	60–100	50–200	25

For each of the three planning horizons (720, 1440 and 2880), for each of the two banner types (horizontal and vertical), we created three sets of problems with three different levels of n, for a total of 18 sets of problems each consisting of 25 problems. The problem set parameters are shown in Table 1.

In order to compare the space utilization of the VF-MaxSpace model to that of the fixed frequency original MaxSpace model, we used the same base set of test problems discussed above. For the fixed frequency problems, we used a display frequency for each problem of $(U_i+L_i)/2$ from the VF-MaxSpace problem. Although each problem has its inherent differences and therefore we really cannot do any paired comparisons at the problem level, this strategy will allow us to compare overall space utilization of the VF-MaxSpace model with the fixed frequency counterpart, the MaxSpace model.

4.2. Computational results

All of the computational studies were performed on a Pentium 4 computer with a 3.6 GHz processor, 1 GB of RAM and a Windows XP® operating system. The proposed heuristics were coded in Visual Basic .Net 2010 and applied to each data set. In this section the results for CPLEX® and the heuristics are provided in order to (i) compare the performance of the proposed VF-MaxSpace model with the traditional MaxSpace model and (ii) to evaluate the relative effectiveness and scalability of the heuristics.

Summarized results for the 18 sets of VF-MaxSpace problems and the 18 sets of MaxSpace problems are presented in Tables 2 and 3. Table 2 presents the results obtained by CPLEX® , while Table 3 presents the heuristic results. The chosen measure of effectiveness is the percent utilization of the available advertising space. If we call the total space available as TS, and the utilized space as US, then the % Utilization=(US/TS) $\times$100. It should be

Table 2
Average percent space utilization and CPU times (using CPLEX®) for the MaxSpace and the VF-MaxSpace models.

Problem set (H:horizontal, V: vertical)	Average percent space utilization			Average CPU Time (s)		
	Fixed-frequency MaxSpace	Variable frequency VF-MaxSpace	Improvement due to variable frequencies	Fixed-frequency MaxSpace	Variable frequency VF-MaxSpace	CPU time limit (s)
720H-1	93.83	99.23	5.40	906	922	1000
720H-2	94.23	99.33	5.10	838	830	1000
720H-3	95.96	99.31	3.35	843	902	1000
720V-1	94.38	99.67	5.29	978	946	1000
720V-2	94.18	99.57	5.39	835	807	1000
720V-3	96.94	99.59	2.65	921	947	1000
1440H-1	92.14	98.76	6.62	3963	3680	4000
1440H-2	93.28	99.26	5.98	2669	2987	4000
1440H-3	97.66	99.06	1.40	3880	3944	4000
1440V-1	93.62	99.35	5.73	3452	3705	4000
1440V-2	93.62	99.41	5.79	3853	2683	4000
1440V-3	97.77	99.52	1.75	3642	3858	4000
2880H-1	91.92	99.28	7.36	15,115	14,905	16,000
2880H-2	91.11	96.32	5.21	15,170	15,064	16,000
2880H-3	96.42	99.38	2.96	15,880	14,776	16,000
2880V-1	96.11	99.76	3.65	15,609	15,817	16,000
2880V-2	95.25	97.81	2.56	15,025	15,061	16,000
2880V-3	97.31	98.77	1.46	15,673	15,881	16,000
Average	**94.76**	**99.08**	**4.31**	**6625**	**6539**	

Table 3
Average percent space utilization and CPU times (using VF-LVMF) for the Maxspace and the VF-MaxSpace models.

Problem set (H:horizontal, V: vertical)	Average percent space utilization			Average CPU time (s)	
	Fixed-frequency MaxSpace	Variable frequency VF-MaxSpace	Improvement in space utilization due to variable frequencies	Fixed-frequency MaxSpace	Variable frequency VF-MaxSpace
720H-1	90.37	98.31	14.52	< 1	< 1
720H-2	82.05	96.57	7.94	< 1	< 1
720H-3	95.31	97.95	2.64	< 1	< 1
720V-1	89.42	98.32	8.90	< 1	< 1
720V-2	89.21	98.16	8.95	< 1	< 1
720V-3	95.77	98.29	2.52	< 1	< 1
1440H-1	93.08	98.02	7.90	< 1	< 1
1440H-2	90.37	98.27	4.94	< 2	< 3
1440H-3	97.01	98.61	1.60	< 2	< 3
1440V-1	88.49	97.65	9.16	< 1	< 1
1440V-2	93.19	98.56	5.37	< 2	< 3
1440V-3	97.45	98.39	0.94	< 2	< 3
2880H-1	95.22	98.36	5.18	< 2	< 4
2880H-2	96.28	98.11	3.14	< 5	< 8
2880H-3	92.19	97.37	1.83	< 6	< 11
2880V-1	94.34	98.44	4.10	< 2	< 4
2880V-2	93.28	98.65	5.37	< 5	< 8
2880V-3	97.09	98.41	1.32	< 6	< 11
Average	**93.41**	**98.14**	**5.35**		

noted that for every problem, the total space demanded was in excess of the quantity of available space, so any unused capacity is due to a scheduling infeasibility or solution deficiency. In Table 2, we report average % space utilization over 25 problems for each of the 18 datasets using CPLEX®. For each problem, we initially planned to allow CPLEX® to run for the length of the given planning horizon (720 min, 1440 min or 2880 min) or until it located an optimal solution. However, we noted that there was virtually no improvement in space utilization beyond approximately 300 CPU seconds for the 720 slot problems, 1000 s for the 1440 slot problems and 4000 s for the 2880 slot problems. Further, we observed that if we did not utilize a CPU limit, the decision tree for many problems became too large and overwhelmed the system's memory capabilities (even when utilizing the memory emphasis parameter within CPLEX®). Based on these observations, we developed a set of CPU time limits that allowed CPLEX® to achieve nearly all of its solution quality improvements while not shutting down the system. We placed a limit of 1000 CPU seconds for the 720 slot problems, 4000 s for the 1440 slot problems and 16,000 s for the 2880 slot problems. These time limits are presented along with the performance results in Table 2. It may be noted that the average CPU time for a particular data set is slightly less than the corresponding CPU time limit. This is because CPLEX® solved some of the problems to optimality in less than the time allowed. The improvement column shows the difference in the percent space utilization between the fixed and the variable frequency models.

The proposed variable frequency model dominated the fixed frequency model for every data set, offering on average an improvement of 4.31 percentage points. Given that the banner advertising industry is a multi-billion dollar industry, a 4.31% increase in revenue is a significant improvement in terms of the monetary benefit to the industry. An added benefit is that variable frequency model also provides more flexible contract options for the publishers and advertisers.

We note that the average improvement in space utilization is lower for the datasets with a larger n (H3 and V3) as a result of the underlying problem structure. Given that the amount of available advertising space is stable across a given planning horizon, as per our earlier discussion in Section 4.1, increasing the number of advertisers (n) required us to simultaneously decrease the upper and lower frequency limits (L_i and U_i) in order to develop realistic yet challenging problems. For the H3 and V3 datasets, which have lower values of the frequency bounds, the relative individual volume requirements for each customer are lower. The lower per capita volume requirements and the increased number of potential advertisements allow for tighter packing (or high space utilization) even for the fixed-frequency model, leaving less room for improvement for the variable frequency model.

In Table 3, we report the percent space utilization and CPU times for the heurstics, for both the fixed frequency and the variable frequency problems. We performed experiments using all four fixed frequency heuristics, viz., LVMF, LVLF, SVMF and SVLF for the fixed frequency problems, and all four variable frequency heuristics for the variable frequency problems. In our testing, we found that the LVMF and VF-LVMF heuristics dominated all the other heuristics; therefore, in the interest of space we only report the results of these two. Consistent with the results in Table 2, results in Table 3 also reveal that the variable frequency model gives significantly higher space utilization than the fixed frequency model, an average of 5.35 percentage points over all 18 datasets. Given the planning horizon duration of the tested datasets, in most cases the utilization of CPLEX® or a similar software package should be feasible. However, in some situations such as a late addition of advertisements to the corpus, an acceptable ad display schedule must be developed in a matter of minutes instead of hours. The described heuristics will be very useful in those situations. It should be noted that one of the key determinants of problem complexity is the number of ad sizes offered by the publisher. In this work, we attempted to model a situation that is common for most publishers within which the publisher offers a relatively wide variety of ad size choices. These problems are quite complex and therefore took quite a bit of time for CPLEX® to find a good solution. In preliminary testing, we solved the models using only two or three ad size choices for each banner configuration with a limited number of advertisers (approximately half as many as the current H1 and V1 datasets) and found that CPLEX® found a near optimal solution in a matter of minutes for even the longer planning horizons. Publishers that have fewer clients and have chosen this more limited offering should not need alternative solution techniques.

5. Summary, conclusions and future research

In this paper, we propose a new model variation to the NP-hard online ad-scheduling problem. We propose the relaxation of the fixed display frequency constraint, which is used in current models. Instead of using a fixed frequency for each ad, we introduce a variable frequency that allows the ad to be served between a lower and an upper bound for a given planning period. This new model will provide publishers an alternative that will expand the flexibility of their contract options while simultaneously improving revenues. In addition, we also utilize the industry accepted standardized banner sizes and ad sizes when developing our test problems. For each of the problems, we proposed and tested several alternative solution techniques. The proposed model and the traditional Max-Space model were each tested for both horizontal and vertical banners, for planning periods of 720, 1440 and 2880 time slots. The main contribution of the paper is the development of an improved ad scheduling model that takes into account real-world issues related to banner advertising, which have been ignored in literature. This contribution is reflected in (i) the formulation of the VF-MaxSpace problem, (ii) proposed heuristics (VF-LVMF, VF-LVLF, VF-SVMF and VF-SVLF) that take advantage of the lower and upper bound frequencies and (iii) the use of test problems that involve standard ad sizes.

The proposed model dominated the traditional model in terms of solution quality. Incorporation of this model could result in increased flexibility and improved revenues for online advertisement publishing agencies. For example, our empirical results found an improvement in revenues of 4.3% on average across all of the datasets. In addition, we developed and tested several heuristic approaches that could be very useful when employees of an online publishing agency find themselves in a time crunch in terms of the amount of time that they have to develop their ad schedule. The VF-LVMF heuristic (Largest Volume, Most Full) provided the best results among all heuristics tested for both horizontal and vertical banners.

This study was focused on banner ad scheduling with linear pricing and did not consider the possibility of volume-based pricing discounts. Future research may consider the inclusion of such a discount pricing schedule strategy, the incorporation of other advertisement types and/or the development of alternative solution approaches.

Appendix A

For a small problem we will show how the LVMF and VF-LVMF heuristics work. We solve a vertical banner problem.

Example problem (vertical banner problem)

Say $N=10$, i.e. there are 10 slots.
Let $n=8$, i.e. eight advertisers.
The height of each banner is 600.

Let the lower and the upper bound demands for each advertiser be as follows:

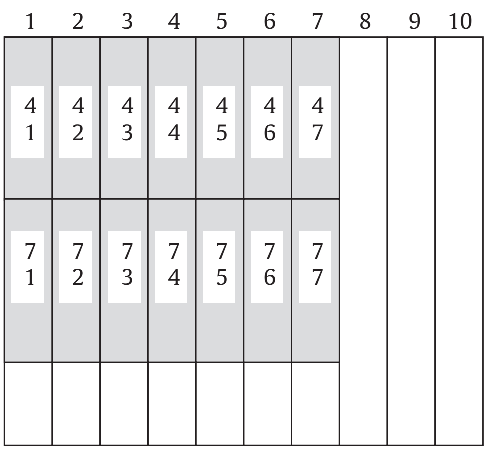
Advertiser i	Ad size s_i	L_i	U_i	$Vol_i=U_is_i$
1	60	3	7	420
2	90	2	4	360
3	240	2	4	960
4	240	6	7	1680
5	90	4	5	450
6	90	4	4	360
7	240	2	7	1680
8	60	2	6	360

The LVMF heuristic

1. Sort the ads by volume in descending order:

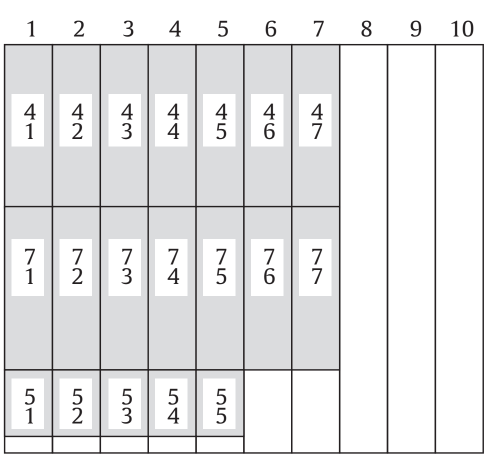
Advertiser i	Ad size s_i	L_i	U_i	$Vol_i = U_i s_i$
4	240	6	7	1680
7	240	2	7	1680
3	240	2	4	960
5	90	4	5	450
1	60	3	7	420
2	90	2	4	360
6	90	4	4	360
8	60	2	6	360

2. The first ad in the sorted list is ad #4 of size 240 with an upper bound frequency of 7. Assign seven impressions of size 240 in slots 1–7.
3. The next ad in the list is ad #7 with upper bound frequency of 7. Assign seven impressions of size 240 each in slots 1–7 since they are the most full. At this point, the slots look like this:



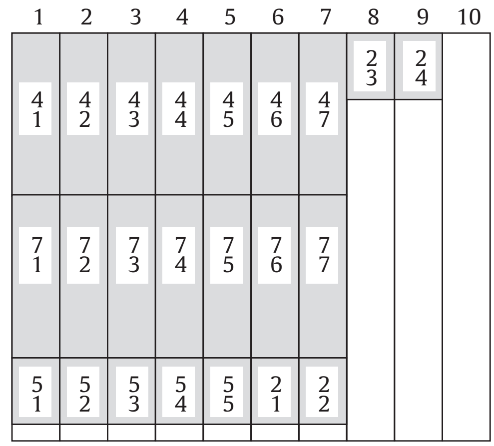
At this point, the remaining capacity in the first seven slots is 120 each. The remaining capacity in slots 8–10 is 600.

4. The next ad in the sorted list is ad #3 of size 240 with upper bound frequency of 4. Only three slots are available that can accommodate a size 240 ad impression, so ad #3 is infeasible.
5. The next ad in the list is ad #5 of size 90 with upper bound frequency of 5. Assign all five impressions to slots 1–5 since they are the most-full slots.



The remaining capacity in slots 1–5 is 30, in 6 and 7 is 120 and in 8–10 is 600.

6. The next ad in the list is ad #1 of size 60 and upper bound frequency of 7. Since there is no room for these seven ads, we do not assign ad #1.
7. The next ad in the list is ad # 2 of size 90 and upper bound frequency of 4. Since there is room for these four impressions, we assign them. The first two impressions go in slots 6 and 7 and the remaining two go in slots 8 and 9.



The remaining capacity in slots 1–7 is 30, in slots 8 and 9 is 510 and in slot 10 is 600.

8. The next ad in the list is #6 of size 90 and upper bound frequency of 4. Since there is no room for four impressions, assigning this ad is infeasible.
9. The next two ads in the list (#8 and #1) both have an upper bound greater than three and are therefore infeasible.

So the assignment is complete and the space utilized is $240 \times 7 + 240 \times 7 + 90 \times 5 + 90 \times 4 = 4170$.

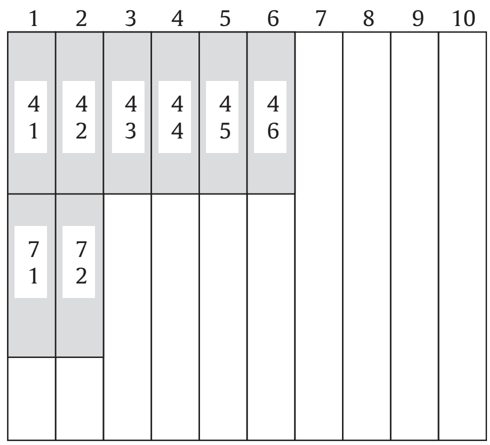
The VF-LVMF Heuristic

1. Sort the ads by volume in descending order:

Advertiser i	Ad size s_i	L_i	U_i	$Vol_i = U_i s_i$
4	240	6	7	1680
7	240	2	7	1680
3	240	2	4	960
5	90	4	5	450
1	60	3	7	420
2	90	2	4	360
6	90	4	4	360
8	60	2	6	360

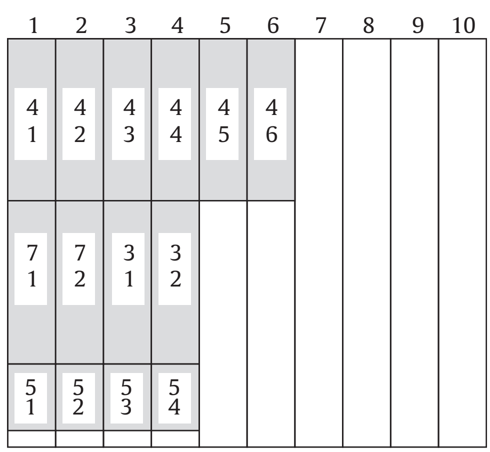
2. The first ad in the sorted list is ad #4 of size 240 with a lower bound frequency of 6. Assign six impressions of size 240 in slots 1–6.
3. The next ad in the list is ad #7 with lower bound frequency of 2. Assign two impressions of size 240 each in slots 1 and 2 since they are the most full.

At this point, the slots look like this:



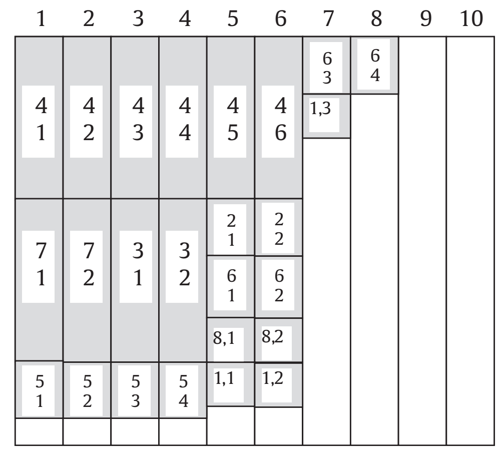
The remaining capacities are: 1 and 2: 120; 3–6: 360; 7–10: 600.

4. The next ad in the sorted list is ad #3 of size 240 with lower bound frequency of 2. Assign two impressions of size 240 in slots 3 and 4.
5. The next ad in the list is ad #5 of size 90 with lower bound frequency of 4. Assign four impressions of size 90 in slots 1–4 since they are the most full. At this point, the slots look like this:



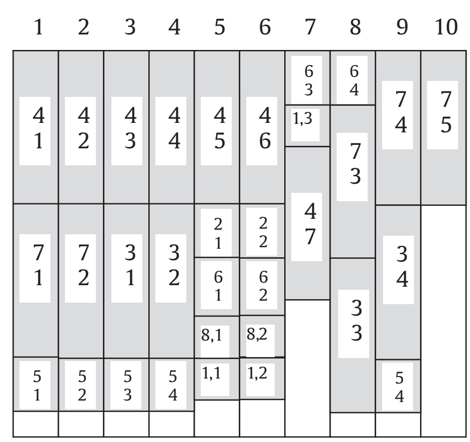
At this point the remaining capacities are: 1–4: 30; 5 and 6: 360; 7–9: 600

6. The next ad in the list is ad #1 of size 60 with lower bound frequency of 3. They can be assigned to slots 5–7.
7. The next ad in the list is ad # 2 of size 90 and lower bound frequency of 2. Since there is room for these two impressions, we assign them to slots 5 and 6.
8. The next ad in the list is #6 of size 90 and lower bound frequency of 4. Since there is room for three impressions, we assign them to slots 5–8.
9. Next, 2 impressions of ad #8 of size 60 are assigned to slots 5 and 6.



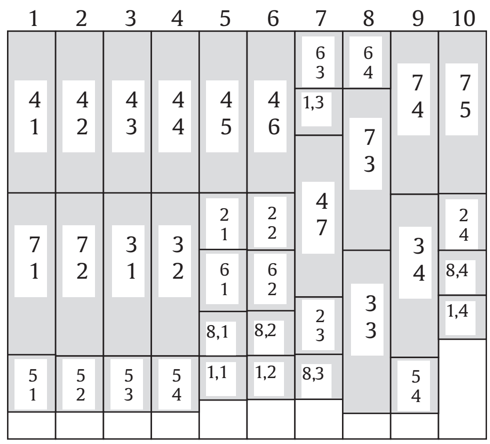
At this point all the lower bound frequencies that were feasible have been assigned. The remaining capacities are slots 1–4: 30, slots 5 and 6: 60; slot 7: 480; slots 8–10: 600. Now we assign ads up to their upper bounds if possible.

10. The first ad in the list is #4 and U_4 is 7, so one more impression for ad #4 needs to be assigned. It is assigned to slot 7.
11. The next ad in the list is #7 and U_7 is 7, so five more impressions need to be assigned but there is room for only three of them, which are assigned to slots 8–10.
12. The next ad in the list is # 3 and U_3 is 4, so two more impressions need to be assigned. They are assigned to slots 8 and 9.
13. The next ad in the list is #5 and U_5 is 5, so one impression needs to be assigned, which is assigned to slot 9. At this point the slots look like this:



14. Next, two impressions of ad #2 are assigned to slots 7 and 10.
15. Next, ad #8 needs four spaces, but has only two available in slots 7 and 10.
16. Finally there is room for one impression for ad #1 in slot 10.

The final schedule looks like this:



The value of this schedule is 5490.

References

[1] IAB. Internet advertising revenues in Q3 '08 at nearly $5.9 billion. URL: 〈http://www.iab.net/about_the_iab/recent_press_releases/press_release_archive/press_release/pr-041311〉 [2010].

[2] IAB. Internet advertising revenue report. URL: 〈http://www.iab.net/about_the_iab/recent_press_releases/press_release_archive/press_release/pr-052611〉 [2010].

[3] Schneider GP. Electronic commerce. Course technology. Cengage Learning; 2010. Isbn:13:978-0-538-46924-1.

[4] Adler M, Gibbons PB, Matias Y. Scheduling space-sharing for Internet advertising. Journal of Scheduling 2002;5(2):103–19.

[5] Dawande M, Kumar S, Sriskandarajah C. Performance bounds of algorithms for scheduling advertisements on a web page. Journal of Scheduling 2003;6(4):373–93.

[6] Kumar S, Jacob VS, Sriskandarajah C. Scheduling advertisements on a web page to maximize revenue. European Journal of Operations Research 2006;173:1067–89.

[7] Kumar S, Dawande M, Mookerjee V. Optimal scheduling and placement of Internet banner advertisements. IEEE Transactions on Knowledge and Data Engineering 2007;19:1571–84.

[8] Freund A, Naor J. Approximating the advertisement placement problem. Journal of Scheduling 2004;7:365–74.

[9] Amiri A, Menon S. Scheduling banner advertisements on the web. INFORMS Journal on Computing 2004;16(1):95–105.

[10] Amiri A, Menon S. Scheduling banner advertisements with multiple display frequencies. IEEE Transactions On Systems, Man, And Cybernetics 2006;36(2):245–51.

[11] IAB. Ad unit guidelines. URL: 〈http://www.iab.net/iab_products_and_industry_services/1421/1443/1452〉 [accessed June 19, 2010].

[12] Balas E, Martin C. Pivot and complement-heuristic for 0–1 programming. Management Science 1980;26(1):86–96.

[13] Chu P, Beasley JA. Genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics 1998;4:63–86.

[14] Freville A, Plateau G. An efficient preprocessing procedure for the multidimensional 0–1 knapsack problem. Discrete Applied Mathematics 1994;49:189–212.

[15] Hübner A, Kuhn H. Retail category management: state-of-the-art review of quantitative research and software applications in assortment and shelf space management. Omega 2012;40(2):199–209.

[16] Bertsimas D, Demir R. An approximate dynamic programming approach to multidimensional knapsack problems. Management Science 2002;48(4):550–65.

[17] Gilmore PC, Gomory RE. The theory and computation of knapsack functions. Operations Research 1966;14:1045–75.

[18] Vasquez M, Vimont Y. Improved results on the 0–1 multidimensional knapsack problem. European Journal of Operational Research 2005;165(1):70–81.

[19] Pirkul H. Heuristic solution procedure for the multiconstraint zero–one knapsack problem. Naval Research Logistics 1987;34(2):161–72.

[20] Gavish B, Pirkul H. Efficient algorithms for solving multiconstraint zero–one knapsack problems to optimality. Mathematical Programming 1985;31(1):78–105.

[21] Eilon S. Application of the knapsack model for budgeting. Omega 1987;15(6):489–94.

[22] Albareda-Sambola M, Fernańdez E, Saldanha-da-Gama F. The facility location problem with Bernoulli demands. Omega 2011;39:335–45.

[23] Syam S, Côté M. A location-allocation model for service providers with application to not-for-profit health care organizations. Omega 2010;38(3):157–66.

[24] Albareda-Sambola M, Fernańdez E, Laporte G. Heuristic and lower bounds for a stochastic location routing problem. European Journal of Operational Research 2007;179:940–55.

[25] Pelegrín B, Fernández P, Pérez M, Hernández S. On the location of new facilities for chain expansion under delivered pricing. Omega 2012;40(2):149–58.

[26] Adenso-Diaz A, Rodriguez F. A simple search heuristic for the MLCP: application to the location of ambulance bases in a rural region. Omega 1997;25(2):181–6.

[27] Costa A, França P, Filho C. Two-level network design with intermediate facilities: an application to electrical distribution systems. Omega 2011;39(1):
3–13.

[28] Redondo J, Fernández J, Arrondo A, García I, Ortigosa P. Fixed or variable demand? Does it matter when locating a facility? Omega 2012;40(1):9–20.