# SCHEDULING ONLINE ADVERTISEMENTS TO MAXIMIZE REVENUE UNDER NON-LINEAR PRICING

**JASON DEANE**
Virginia Tech
Virginia

**ANURAG AGARWAL**
University of South Florida
Florida



**ABSTRACT**

*Internet popularity is at an all time high. This phenomenon continues to drive enormous revenue growth for the online advertising industry. One of the difficult challenges facing this industry is determining how to schedule online advertisements. We extend prior research in this area by developing a quantitative model which incorporates a very important economic strategy, volume pricing discounts. In addition, motivated by the NP-Hard nature of the problem, we propose and test a several heuristic and metaheuristic solution approaches. We show that explicit consideration of this economic pricing strategy in the scheduling process can, under certain elasticity of demand assumptions, significantly improve the revenue generation ability of online ad publishers. This provides significant support for the proposed model.*

**Key Words:** Scheduling, Pricing Discounts, Optimization, Heuristics, Non-Linear Pricing

## 1. INTRODUCTION AND BACKGROUND

According to a recent Price Waterhouse Coopers/Interactive Advertising Bureau (IAB) report, revenues for the online advertising industry in 2011 was $ 31 billion, which represented an approximately 22% increase over prior year revenues [9]. Further, online advertisement revenues have been growing at an average of roughly 20% per year for the past several years. Online advertisement revenues are also growing as a proportion of all advertisement revenues from all channels (including television, radio and newspaper). In 2011, online advertisement revenues represented almost 25% of all advertising revenues, up from approximately 19% in 2009 [9]. This growing trend is expected to continue in the future as Internet popularity pushes companies to channel more of their advertising dollars toward the online medium. Online advertisements take several different forms including search based text ads, classified and directory ads, rich media, banner ads, digital video commercials, pop ups etc. Search based text ads account for roughly 45% of all online ad revenues. Banner ads are rank second in popularity of use, accounting for approximately 22% of all online ad revenues. For example in 2011, banner ad revenues accounted for 6.8 billion dollars (approximately 22% of the total industry revenues of 31 billion).

Developing a revenue optimizing banner advertisement schedule given the limited advertising space available is a challenging problem which has received considerable attention in the literature [1, 2, 4, 5, 6, 10, 11]. Adler et al. [1] first proposed the MaxSpace model, which is essentially an integer programming formulation to schedule banner ads to maximize the utilization of available space. This model, given their linear pricing assumption, also maximizes revenue. Adler et al. also assumed a fixed display frequency display, i.e. either all or none of the requested frequency of ads, for a given advertisement, are displayed. Dawande et al. [4] and Kumar et al. [10, 11] discussed various complexity issues associated with the MaxSpace problem and showed that the problem is NP-Hard in nature. In addition, they proposed Genetic Algorithms to solve the problem. Amiri and Menon [2] extended the original MaxSpace problem to include multiple display frequencies as opposed to fixed display frequencies. Deane and Agarwal [6] further extended the work to incorporate variable display frequencies which incorporated for each ad a lower and an upper bound of display. In addition, they demonstrated that by using variable display frequencies, the space utilization and hence the revenue therefrom is higher. They call their model the VF-MaxSpace model. Deane [5] developed a model that incorporates contextual ad targeting. Regardless of the volume of advertisements requested for a particular ad, all these proposed models assume a linear pricing structure. This limits their applicability in industry as most companies offer volume based discounts in an effort to improve their ability to compete for the high volume advertisers.

Major online advertising companies, such as Google, Yahoo, AOL and Facebook have adopted business models largely driven by the generation of revenue through their online advertisement publishing efforts. These companies are constantly competing for the same advertising customers. One way organizations commonly compete in any competitive industry is through their pricing strategies, particularly through offering quantity discounts to attract large volume customers. Yet the current models in the literature on online advertisement scheduling do not address such non-linear pricing strategies and are hence deficient in their applicability to the real-world scheduling needs of online advertising companies. In this study we fill this gap by extending Deane and Agarwal's [6] VF-MaxSpace model to incorporate a non-linear objective function to incorporate quantity discounts.

In Section 2, we discuss the economics of the quantity discount pricing strategy to motivate the proposed non-linear ad scheduling model. In Section 3, we describe the details of the formulation of our proposed model. We call our model the NLP-VF-MaxSpace model, or just NLP-VF for short, where NLP stands for nonlinear pricing and VF stands for variable frequency. In Section 4, we describe several solution approaches, including a new heuristic that explicitly incorporates the volume discount cutoff points. We also describe a Genetic-Algorithms based approach that we use to solve the proposed problem. In Section 5, we describe our empirical design to test the efficacy of the proposed model. We also provide and discuss the experimental results. In Section 6, we provide a discussion of the important managerial implications and suggest ideas for future related research.

## 2. ECONOMICS OF THE NON-LINEAR PRICING MODEL

Under a linear pricing structure, the advertiser (or the customer) is charged the same amount per unit of advertising space regardless of volume. In the industry, this is not commonly the case, as companies routinely offer quantity discount breaks to attract larger volume customers or to incite customers to purchase more advertising space. These price breaks are commonly implemented through a step-pricing function under which the per-unit price for additional display of ads is discounted at each step. As an example, suppose from a volume of 1 impression to the first cutoff display frequency of say 100 units, the rate is 50 cents per unit, from the first cutoff to the second cutoff of say 120 units, suppose the rate is 45 cents per unit and from the second to the third cutoff of say 140 units, say the rate is 40 cents per unit, and so on. For this example, the revenue for 100 units would be \$50; the revenue for 110 units would be \$54.50 (\$50 for the first 100 units plus \$4.50 for the next 10 units); the revenue for 120 units would be \$59 (\$50 + \$9); the revenue for 130 units would be \$63 (\$50 for the first 100 + \$9 for the next 20 + \$4 for the next 10) and so on. Under a linear-pricing scheme, assuming sufficient customer demand, revenue for 130 units would be \$65. However, under the linear-pricing scheme, an advertiser may choose to only purchase 100 units, generating revenue of only \$50, whereas the quantity discount may incite the customer to purchase extra units to take advantage of the reduced prices resulting in higher revenues. The amount of extra space that the customer purchases depends of course on the price elasticity of the demand for ad space. The actual cutoffs and discounts will differ from publisher to publisher and from time to time and will be governed to some extent by the competition in the market. The proposed non-linear pricing structure assists publishers in their efforts to increase market share and maximize revenue while providing additional pricing options for the advertisers.

It should be noted that the nonlinear-pricing variable frequency (NLP-VF) model is meaningful only under certain economic conditions. If the ad-space demand at the regular market rate, i.e., with no quantity discount, exceeds the publisher's available ad space, then offering quantity discounts makes little economic sense as it would reduce revenues. Under such a situation, a linear pricing model such as the VF-Maxspace model [6] would be applicable and sufficient. Conversely, if the quantity of ad space demanded at the prevailing market rate is less than the available space, i.e. if there is excess capacity, then all of the customer's demand for ads can be accommodated without the need for any optimization model. However, in this situation, the publisher would obviously consider using discount pricing strategies in order to boost the demand for ad space and consequently profits. If the discount rate offered is such that the new demand for ad-space exceeds the available capacity then the NLP-VF model is applicable. If the available ad-space is still in surplus in spite of the offered pricing discounts, then no optimization model is necessary. However, market forces always determine market rates in a competitive environment such that the quantity demanded roughly matches the quantity supplied. Thus, given the observed elasticity of demand for banner-ad space, the publisher can always offer discounts such that the new demand exceeds the available capacity, creating the conditions necessary for the usefulness of the NLP-VF model.

## 3. THE PROPOSED NON-LINEAR MODEL

Before providing the IP formulation of the proposed model we discuss, at a high level, how online ad scheduling process for the proposed situation. A web site, upon which ads are scheduled, has


**FIGURE 1:** A screen print of techsideline.com web page. Notice the advertising banner down the right hand side of the webpage.

either a vertical banner or a horizontal banner or both, dedicated for ad display. See Figure 1 for an example of a web site with a vertical banner. The advertisements displayed in a given banner are refreshed/changed each minute. In other words, the time dimension is divided into slots of one-minute each and ads are assigned to these 1-minute time slots. Advertisers are charged per display. Therefore it is up to the advertiser to provide the ad publisher with some guidance as to how many times they wish to have their ads displayed during a pre-defined planning period. This work assumes a variable display frequency which provides some flexibility for the ad publisher. An advertiser, based on their perception of what it will take to achieve an acceptable level of market penetration, provides the publisher with an upper and a lower bound of display frequencies for each ad for the chosen planning horizon. If the publisher chooses to display this ad, the number of impressions must fall between these pre-defined bounds and the advertiser is charged for the exact number of displayed impressions. The publisher defines a planning horizon for which ads are scheduled in advance. For example, if the planning horizon is one day, there will be 1,440 time slots to be filled as there are 1,440 minutes in a day.

There are several assumptions inherent in the formulation. First, it is assumed that each banner/time slot has the same height (for vertical banners) and width (for horizontal banners) of ad space dedicated for displaying ads. This height (or width) is denoted by $S$. This is a very realistic assumption because the designers of the web site designate a predetermined part of the web page for displaying banner ads. These dimensions will typically be coded in the html file or the style sheets of the web page and therefore they are likely to remain constant throughout the day. Second, we assume that only one ad fits within the width (for vertical banners) or height (for horizontal banners), i.e. two (or more) ads cannot fit side-by-side on a vertical banner. This is also a very realistic assumption as the banner ad sizes are of standardized widths and typical ad sizes are such that only one ad will fit on the width of a vertical banner. Note that several ads will fit along the height of a vertical banner (or along the width of a horizontal banner). However, it is certainly plausible for an ad publisher to allow ads of such sizes that multiple ads may be scheduled in both directions. Our model will not work in this situation. Third, each advertisement has a height that is less than or equal to the height (or width in case of horizontal banners) of the banner, $S$. Clearly if the height of a vertical banner exceeds $S$, it cannot be assigned to that website.

We will now describe our non-linear variable frequency MaxSpace problem (NLP-VF-MaxSpace). As previously stated, this model is an extension of the VF-MaxSpace problem originally proposed by Deane and Agarwal [6]. The mathematical formulation of NLP-VF-MaxSpace problem is as follows:

$$Max \sum_{i=1}^{n} f_i\left(\sum_{j=1}^{N} s_i x_{ij}\right) \quad (1)$$

s.t.

$$\sum_{i=1}^{n} s_i x_{ij} \leq S, \; j = 1,2,\dots,N \quad (2)$$

$$L_i y_i \leq \sum_{j=1}^{N} x_{ij} \leq U_i y_i, \; i = 1,2,\dots,n \quad (3)$$

$$x_{ij} = \begin{cases} 1 \; if \; ad \; i \; assgined \; to \; slot \; j \\ 0 \; otherwise \end{cases} \quad (4)$$

$$y_i = \begin{cases} 1 \; if \; ad \; i \; is \; assgined \\ 0 \; otherwise \end{cases} \quad (5)$$

Where

$n$ : number of advertisements

$N$ : number of Banner/Time slots

$S$ : Banner height

$s_i$ : height of advertisement $i$, $i = 1,2,\dots,n$

$L_i$ : lower bound on the frequency of advertisement $i$, $i = 1,2,\dots,n$

$U_i$ : upper bound on the frequency of advertisement $i$, $i = 1,2,\dots,n$

$f_i()$ : the non-linear step function of price per unit volume, $e$ $i$

The objective of the publisher, stated in (1), is to maximize the amount of revenue generated through ad publishing for a given set of banner slots. Please note that the objective function uses a function $f()$, which is a function of the product of total frequency and the size of the ad displayed. This function is a step function and hence non-linear, although it is step-wise linear. At each step, the pricing is different depending on the volume. 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 for horizontal banners). 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.

The primary difference between NLP-VF-MaxSpace model and the VF-MaxSpace model [6] is the pricing function $f()$ in the objective. The NLP-VF-MaxSpace model attempts to maximize the dollar-value revenue from the displayed ads whereas the VF-MaxSpace problem attempts to maximize the usage of ad space. The main motivation for developing this model is to help the publishers maximize their profits given a step pricing function. The determination of profit-maximizing non-linear pricing function depends on the market conditions and in particular on the price elasticity of demand. As market conditions change, the pricing function needs to be updated. In a perfectly competitive market, the pricing function will be determined by the markets and the publisher will have to use the pricing function as a given and then rely on optimization models to maximize the revenues, given the market-determined pricing function. In non-competitive environments, the publisher will come up with the pricing function through trial and error because the exact price elasticity of demand is hard to measure. The proposed model can be used to help the publisher determine the possible revenues for different pricing functions.

Due to the non-linearity of the objective function, the NLP-VF-MaxSpace problem cannot be solved using the Simplex method, and linearizations or the use of non-linear solvers would lead to unacceptable solution times for this NP-Hard problem. In order to solve problems of this nature, we therefore need heuristic and metaheuristic approaches, even for smaller problems.

## 4. MODEL SOLUTION APPROACHES

Deane and Agarwal [6] show that the variable frequency advertisement scheduling problem is NP-hard. The NP-hard nature makes it highly unlikely that it will ever be solved by an efficient optimal algorithm [7]. Therefore, efficient and effective approximation algorithms or heuristics are necessary. For the VF-MaxSpace problem, the best heuristic is the VF-LVMF (Variable-Frequency-Largest-Volume-Most-Full). For the NLP-VF problem, we propose a heuristic that takes into account the quantity discount cutoffs. We call this heuristic the NLP-VF-LVMF heuristic. To test the efficacy of this heuristic, we will compare its performance with that of the VF-LVLF heuristic. In addition, we propose and describe a Genetic Algorithm approach for solving the proposed problem.

We will describe the VF-LVMF and the NLP-VF-LVMF heuristics next and the genetic algorithm approach next.

### 4.1 The VF-LVMF heuristic

*Variable-Frequency —*
*Largest Volume Most Full (VF-LVMF) Heuristic*

1. For each ad $i$, define $Vol_i = p_m * s_i$ ;
2. Sort the ads in descending order of $Vol_i$
3. Sort slots in the order of occupied volume, most full to least full
4. For each ad $i$ in the sorted list,
   - o 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.
   - o If feasible, assign an impression of ad $i$ to each of the next $L_i$ most-full slots
   - o Sort slots in the order of volume occupied, most full to least full
5. For each ad $i$ in the sorted list,
   - o Assign impressions of ad $i$ to slots until either no space is available or $p_m$ number of impressions are assigned.

### 4.2 The NLP-VF-LVMF heuristic

In order to simplify the explanation of the NLP-VF-LVMF heuristic, without loss of generality, we assume that there are only two discount cutoff points. The NLP heuristic attempts to maximize revenue by iteratively assigning just enough impressions of the chosen ad to move to the next pricing cutoff point. Only after every ad has either had enough impressions scheduled to get to this cutoff point or reached its upper bound does the heuristic attempt to schedule impressions at the next pricing level. Even if there is room to accommodate more impressions of a given ad, it delays additional assignment of that ad until all ads are filled up to the first cutoff point, thus providing maximum revenue for the publisher.

Step 1: Define $Vol_i$ as $p_m * s_i$ for each advertiser; Sort the ads in descending order of *Vol*.

Step 2: Consider each ad in the sorted order. If $L_i$ is less than or equal to the first discount cutoff point and if feasible, i.e., $L_i$ number of slots of size $s_i$ are available, then assign $L_i$ impression of that ad.

Step 3: Consider each ad in the sorted order again. If the lower frequency bound $L_i$ is between the first and the second cutoff points and if it is feasible, i.e., if $L_i$ number of slots of size $s_i$ are available, then assign that ad up to the lower frequency bound.

Step 4: Next consider each ad in the sorted order again. If the lower frequency bound $L_i$ is above the second cutoff point and is feasible, i.e., if $L_i$ number of slots of size $s_i$ are available, then assign that ad up to the lower frequency bound.

Step 5: Consider ads assigned in Step 2 in descending order of their remaining volume and assign each ad up to its upper bound $U_i$ or until no more space is available.

Step 6: Repeat step 5 for the ads assigned in Step-4.

Step 7: Repeat step 5 for the ads assigned in Step-5.

Step 8: Compute the value (or revenue) using the objective function for the LP-VF problem.

### 4.3 Genetic Algorithm (GA)

We also employ a genetic algorithm (GA) based algorithm which was originally introduced by Kumar et al. [10] and successfully utilized by Deane [5]. For the three proposed problems, each GA chromosome, which can be visualized as a 1 x n vector as depicted below, represents a candidate sequence of $n$ advertisements $A = \{a_1, a_2, ..., a_n\}$.

| | | | | |
|---|---|---|---|---|
| $a_2$ | $a_4$ | $a_1$ | $a_5$ | $a_3$ |

The advertisements are served in the order in which they appear in the respective chromosome. For example, given the basic five advertisement chromosome string depicted above, the GA would first attempt to serve advertisement 2, followed by advertisement 4, 1 and so on. When attempting to serve a given ad, if there are not at least $L_i$ (lower frequency bound for advertisement $i$) time slots with sufficient capacity to accommodate the ad, it is not served at all. Those advertisements which do meet this feasibility requirement are served until either their upper frequency bound is reached or an attempt has been made to place the respective advertisement in each time slot. The associated fitness value is measured based on the objective function of the given problem. The three primary operations of a simple genetic algorithm are reproduction, mutation and crossover. We employ a roulette wheel reproduction method, a one-point crossover and a basic advertisement swap mutation operator. Please see appendix A for a more detailed description of the GA operators.

<u>GA Notation</u>

$e$ : elite list percentage　　$p_m$ : probability of mutation
$ps$ : population size　　NU: num of desired unique solutions
CL: crossover attempt limit

The GA begins with an initial population of chromosome strings which are all created randomly with the exception of one string which is created using the appropriate greedy heuristic. Between generations we use the elite list percentage ($e$) to determine how many of the most fit strings will survive unchanged into the next generation. The remaining strings are developed via crossover; therefore, the crossover rate is (1-$e$). The roulette wheel reproduction operator selects potential reproductive parental strings based on their relative fitness values. Each string has a probability of selection which is directly proportional to the ratio of its fitness value divided by the sum of the fitness values of the entire population. The 'most fit' strings are thereby given the highest probability of selection. Given the binary nature of advertisement selection in each of the proposed advertisement scheduling problems, any advertisement duplication within a proposed solution string causes it to be infeasible. As a result, common GA selection and crossover mechanisms struggle to achieve an acceptable level of feasibility for these problems. To overcome this challenge, we use a crossover mechanism developed by Kumar et al. [10] as described below which insures the feasibility of each new offspring. Having selected two parent strings via the roulette wheel process described above, a single crossover point is randomly selected. In the example depicted in Figure 2, point number five, which falls between advertisements five and six, was selected as the crossover point. Based on the chosen crossover point and the genetic material of the parents, two child strings are created. The genetic material on the left side of the crossover point in parent 1 is then directly inherited by child 1 and similarly for parent 2 and child 2. In our example (see Figure 3), the first set of advertisements which are inherited by child 1 are advertisements $a_7$, $a_4$, $a_{11}$, $a_5$ and $a_8$. Up to this point, the proposed crossover method has followed the basic single point crossover process; however, the remainder of the process is somewhat different. Unlike the traditional mechanism, the second half of the genetic material which makes up the chromosome string of child 1 is not directly inherited from parent 2. Instead, the advertisements which make up the second half of child 1's string are inherited from the second half of parent 1 with the difference being that they are reordered based on how they appear in parent 2. A similar process is followed for child 2. In our basic example, the advertisements which make up the second half of child 1 are advertisements $a_9$, $a_{10}$, $a_3$, $a_6$, $a_1$ and $a_2$, but they are reordered based on how they appear in parent 2 (ie. $a_2$, $a_1$, $a_{10}$, $a_3$, $a_9$ and $a_6$). This reproduction process has created two new offspring for the next generation. However, before being added into the next population, the new offspring are given an opportunity to mutate based on the pre-defined probability of mutation operator ($p_m$). A string which is selected for mutation will have two randomly selected advertisements swap places within the string. In the example below (see Figures 4 and 5), we assume that the second child has been selected for mutation and advertisements $a_8$ and $a_{11}$ have been randomly selected as mutation candidates. This entire process in repeated from generation to generation until a predefined number of unique solutions have been created or the crossover attempt limit has been exceeded.

<u>*GA – Algorithm*</u>

Step 1: Initialize

Step 2: Apply the VF-LVMF greedy heuristic and insert the resulting solution as the first string in the initial GA population. Apply the NLP-VF-LVMF heuristic and insert the resulting solution as the second string in the initial population.

Step 3: Complete the initial population by creating (*ps* – 2) random chromosomes

Step 4: Set t = 1 and c = 0.

Step 5: For each string, attempt to assign each of the advertisements in the order in which it appears in the string. If feasible, assign advertisement *i* to the least full slots one at a time until we either reach a time slot which has insufficient capacity to accept advertisement *i* or the upper frequency bound for advertisement *i*, $p_m$ , is reached. Evaluate the fitness of each string based on the objective function. Check each chromosome for uniqueness. For each unique string, set $t = t + 1$

Step 6: Sort the strings in descending order of their relative fitness function values.

Step 7: Populate the elite list by selecting the best ($e * ps$) strings based on their relative fitness values. These strings are added to the next population. These strings remain in the current population to be considered for crossover in step 8.

Step 8: Utilizing the roulette wheel selection method, select two parent strings for reproduction and cross them over. Set $c = c + 1$.

Step 9: Mutate the resulting children based on the mutation probability.

Step 10: Add the children strings to the next population advertisement test them for uniqueness. For each child that is unique, set $t = t + 1$.

Step 11: If $t$ $e$ NU or $c$ $e$ CL, calculate the fitness value for those strings in the new population and terminate reporting the best solution so far.

Step 12: If the number of chromosomes in the next population $e$ *ps*, remove the latest children that were added to the current population one at a time until the number of chromosomes in the population = *ps* and goto step 5; otherwise, goto step step 8.

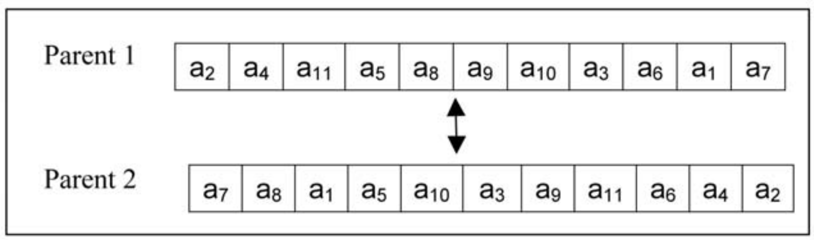


**FIGURE 2:** Parents Prior to Crossover

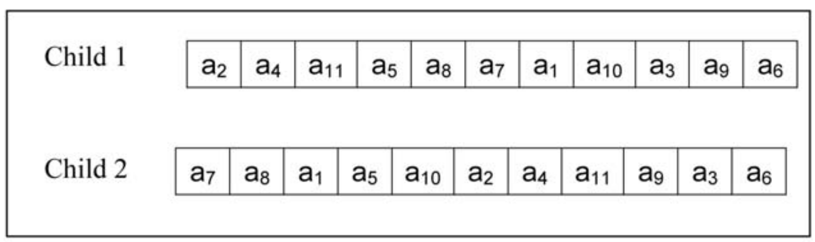


**FIGURE 3:** Offspring

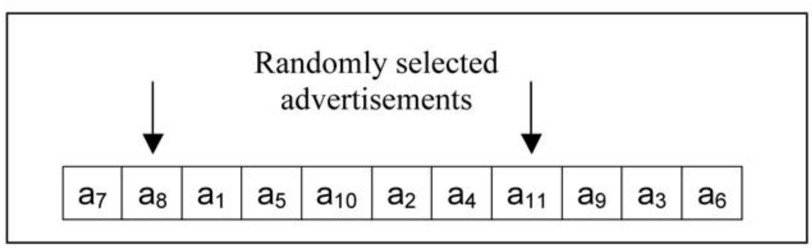


**FIGURE 4:** Child 2, Before Mutation

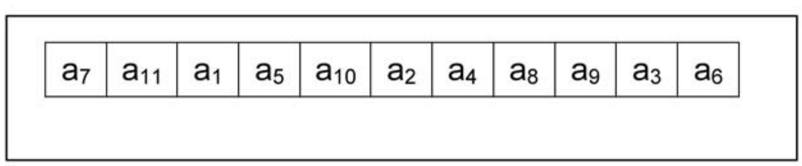


**FIGURE 5:** Child 2, After Mutation

### 4.4 Complexity

The complexity of all the greedy heuristics proposed above is $O(nN \ln(N))$ where $n$ is the number of advertisers and $N$ is the number of slots in the planning period. The heuristics sort both the list of advertisers and the list of slots. A more comprehensive complexity would be $O(n\ln(n)+nN \ln(N))$ but since in general $n$ will be much smaller than $N$, the $n\ln(n)$ term can be ignored for the purpose of figuring complexity. Also, it can be shown that since the VF-Maxspace problem, which has been shown to be NP-Hard [6], is a special case of NLP-VF (when, $f_i$ is the same for all $i$), that NLP-VF is also NP-Hard.

## 5. EXPERIMENTAL TESTING OF PROPOSED HEURISTICS

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

### 5.1 Generation of Test Problems

To test the proposed solution approaches, we generated six sets of 25 problems each for a total of 150 test problems. Three sets are for horizontal banners and three for vertical banners. We use three different planning horizons — 12 hours (720 min), 1 day (1,440 min) and 2 days (2,880 min). See Table 1 for a summary of the six sets of problems. In an effort to align this work with the real-world problems, we use the industry accepted standard sizes developed by the IAB [8]. For example, for horizontal banners, there are two standard sizes — the half banner of size 234x60 pixels and the full banner of size 468x60 pixels. For vertical banners, a total height of 600 pixels is standard, with ad sizes of 120x60, 120x90 and120x240 being standard. Note that 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 on horizontal and vertical banner ads. When generating our problems we made a concerted effort to ensure that the number of ads and their demands ($L_i$ and $U_i$) are such that most of the ads have at least their lower frequency bound ads satisfied. We also made sure that the demand for ad space was not smaller than the capacity, because then an optimization model would not be necessary.

In creating these problems we paid particular attention to the economic justification for these types of problems. Recall from Section 2 that the NLP-VF-MaxSpace problems only make economic sense if the demand for ad space is less than the supply with the standard/regular pricing structure and the demand, at an assumed price elasticity, is greater than the supply when the discounted pricing structure is employed. We ensured that each of the NLP-VF problems met these criteria.

### 5.2 GA Parameter Selection

When running GAs, the values for a number of parameters such as population size, elite percentage, mutation probability etc. have to be chosen. As discussed by Aytug et al. [3], one of the concerns with genetic algorithms is the absence of clear and consistent theoretical guidance as to how best to choose the parameter values. In an effort to avoid the commonly criticized practice of ‘parameter tuning’ and to gain a better understanding of the robustness of each of the proposed techniques for the problems introduced, we maintain a consistent set of parameter settings across all of the problem sets for both problems. The chosen parameter values were selected based on a full factorial design of experiment employed for one NLP-V1 problem sets. We tested the GA for different population sizes (40, 80 and 120), different mutation probabilities (0.01, 0.05 and 0.1) and different elite-list percentage (0.10, 0.25 and 0.40). We tested each of the 27 possible parameter combinations on the set of 25 problems. The combination that provided the best solution quality was a population size of 80, a mutation probability of 0.05 and an elite-list percentage of 0.25 therefore we employ this set of parameter values for our experiments.

### 5.3 Results

We report the results for the two heuristics, i.e. VF-LVMF and NLP-VF-LVMF and the GA, for all six problems set in Table 2. We report the average revenue achieved for each set of problems.

To test if the non-linear pricing model made economic sense, we first report (in the first column of numbers) the revenue that would have been realized under linear pricing, i.e. no quantity discounts and if there was excess capacity. Note that under this situation, all of the demand would be met and no optimization method would be needed to schedule the entire set of ads. We then tried to boost the demand by offering quantity discounts and found that more of the available space was utilized and in fact there was excess demand as a result of the discounts. In order to compute the revenue, we had to assume a certain price-elasticity function. We assumed that a 10% decrease in price would lead to a 10% increase in demand. A more elastic demand, i.e. a higher than 10% increase in demand for a 10% decrease in price would lead to even higher revenues, and conversely a more inelastic demand would lead to a decrease in revenue. We report (in second column of numbers) the realized revenue with non-linear pricing using

**TABLE 1: Description of Problem Sets for the NLP-VF problems**

| Problem Set | Banner Shape | # of Problems | Planning Period (N) | Number of advertisers ($n$) | Lower Bound Frequency ($L_i$) | Upper Bound Frequency ($Ui$) |
|---|---|---|---|---|---|---|
| NLP-H1 | Horiz | 25 | 720 | 40 to 60 | 25 | 60 |
| NLP-H2 | Horiz | 25 | 1,440 | 60 to 80 | 40 | 80 |
| NLP-H3 | Horiz | 25 | 2,880 | 80 to 100 | 60 | 120 |
| NLP-V1 | Vert | 25 | 720 | 40 to 60 | 25 | 60 |
| NLP-V2 | Vert | 25 | 1,440 | 60 to 80 | 40 | 80 |
| NLP-V3 | Vert | 25 | 2,880 | 80 to 100 | 60 | 120 |

**TABLE 2: Average Revenues ($) of NLP-VF Problems**

| | Average Revenue | | | |
|---|---|---|---|---|
| | Linear Pricing. VF-LVMF Heuristic. | Non-Linear Pricing. VF-LVMF Heuristic. | Non-Linear Pricing. NLP-VF-LVMF heuristic. | Non-Linear Pricing. GA |
| NLP-H1 | 288,026 | 300,396 | 301,247 | 302,568 |
| NLP-H2 | 514,170 | 531,525 | 532,478 | 533,037 |
| NLP-H3 | 1,076,805 | 1,112,851 | 1,113,264 | 1,114,125 |
| NLP-V1 | 374,270 | 397,728 | 398,358 | 398,867 |
| NLP-V2 | 727,206 | 804,168 | 806,598 | 807,024 |
| NLP-V3 | 1,478,343 | 1,598,670 | 1,600,790 | 1,601,741 |
| | CPU Time (sec) | | | |
| NLP-H1 | <1 | <1 | <1 | 7.8 |
| NLP-H2 | <1 | <1 | <1 | 25.9 |
| NLP-H3 | 7.28 | 7.89 | 8.12 | 87.9 |
| NLP-V1 | <1 | <1 | <1 | 8.3 |
| NLP-V2 | <1 | <1 | <1 | 27.8 |
| NLP-V3 | 8.24 | 8.58 | 9.02 | 95.2 |



the VF-LVMF heuristic. We note that the revenues increased from $1,478,343 to $1,598,670, an 8.1% increase. This establishes the usefulness of having a non-linear pricing strategy even under a modest elasticity assumption of 10% increase in demand on a 10% decrease in price. Clearly, this 8.1% increase in revenue depends on our assumed elasticity. In reality, the gain in revenue will depend upon the real price elasticity of demand. A higher price elasticity of demand can result in even higher revenues and a more inelastic demand would not benefit so much from a non-linear pricing model. We note that because a non-linear pricing has never been discussed in the literature on online ad scheduling, the issue of price elasticity of demand has also not been discussed in relation to online ad scheduling. We next report (in the third column of numbers) the realized revenue with the quantity discounts using the NLP-VF-LVMF heuristic which attempts to leverage some domain-specific knowledge, specifically the discount cutoff points. The NLP-VF-LVMF heuristic performed slightly better than the VF-LVMF heuristic on average. The revenues increased from $1,598,670 to $1,600,790, an increase of 0.13%. Although as a percentage, an increase of 0.13% might seem insignificant, but for a 22 billion dollar industry, even a 0.13% increase is significant in absolute terms. We tested to see if the NLP-VF-LVMF heuristic was statistically significantly better than the VF-LVMF heuristic using a one-tailed paired t-test and found that the NLP-VF-LVMF heuristic was superior to the VF-LVMF heuristic at the 0.05 significance level. The p-value ranged between 0.013 to 0.0295 for the six sets of problems. In addition, we also tested the GA for the six problem sets and found a marginal improvement (0.06%) over the NLP-VF-LVMF heuristic, at the expense of significantly more CPU time. So the NLP-VF-LVMF heuristic works extremely well for the NLP-VF-MaxSpace model.

## 6. SUMMARY, MANAGERIAL IMPLICATIONS AND FUTURE RESEARCH

In this paper, we fill a void in the current literature by proposing an online ad scheduling model to support a non-linear quantity discount pricing strategy. This represents a real-world variation of the NP-Hard online advertisement scheduling problem. In addition, we propose and test several alternative solution techniques. In an effort to provide potential users with a basic estimation of their performance potential, these techniques were tested against a large set of sample problems which vary in size and difficulty.

Online advertisement publishers currently find themselves in a very competitive industry. One of the primary means of competition is pricing. Firms compete by offering attractive non-linear pricing models. Until now, the online ad-scheduling literature had not provided a supporting scheduling model for this common strategy. In this work, we introduce such a model. The model and the solution approaches proposed in this paper can be used by employees of online ad publishers to (i) schedule ads over a given planning horizon with a nonlinear pricing structure and (ii) (assuming a known elasticity of demand) to develop a set optimal quantity discount rates. In addition to these two organizational benefits, which should help ad publishers improve their operational efficiency, the proposed model will help support decision making for high level managers. The proposed model and solution techniques will provide managers of online ad publishing agencies with a powerful decision support tool by allowing them to compare and contrast different potential pricing strategies for this very challenging scheduling problem. This can provide them some insight as to how structural changes to their ad pricing scheme is likely to impact the bottom line. We are hopeful that these managerial takeaways will provide them with a competitive advantage.

There are several suggestions for future research. First, this and prior studies have focused only on banner ads, which make up only a fraction of all online ads. Future studies should consider the development/incorporation of other types of online ads. Second, this and prior studies have focused on scheduling ads on only a single web site. Future studies should also develop models to schedule ads on multiple sites simultaneously. Third, prior work in online ad scheduling has not taken into account the impacts of traffic congestion. Future studies should explicitly incorporate traffic into the models. We hope that our work will motivate additional research which will be beneficial in our efforts to solve this difficult problem.

## REFERENCES


[1] Adler, M., Gibbons, P.B., and Matias, Y., "Scheduling Space-Sharing for Internet Advertising," *Journal of Scheduling,* 5(2), 2002, 103-119.

[2] Amiri, A. and Menon, S., "Scheduling Banner Advertisements on the Web," *INFORMS Journal on Computing*, 16(1), 2004, 95-105.

[3] Aytug, H., Khouja, M. and Vergara, F.E. "Use of Genetic Algorithms to Solve Production and Operations Management Problems: A Review," *International Journal of Production Research*, 41(17), 2003, 3955-4099.

[4] Dawande, M., Kumar, S. and Sriskandarajah, C., "Performance Bounds of Algorithms for Scheduling Advertisements On A Web Page," *Journal of Scheduling,* 6(4), 2003, 373-393.

[5] Deane, J., "Hybrid Genetic Algorithm and Augmented Neural Network Application for Solving the Online Advertisement Scheduling Problem with Contextual Targeting," *Expert Systems with Applications*, 39, 2012, 5158-5177.

[6] Deane, J. and Agarwal, A., "Scheduling Online Advertisements to Maximize Revenue Under Variable Display Frequency," *OMEGA, The International Journal of Management Science*, 40(5), 2012, 562-570.

[7] Garey, M.R. and Johnson, D.S., "Computers and Intractability: A Guide to the Theory of NP-Completeness," 1979, New York, NY: W.H. Freeman & Co.

[8] IAB Display Advertising Guidelines. http://www.iab.net/guidelines/508676/508767/displayguidelines

[9] IAB (2012) Internet Ad Revenues Hit $31 Billion in 2011, Historic High Up 22% Over 2010 Record-Breaking Numbers http://www.iab.net/about_the_iab/recent_press_releases/press_release_archive/press_release/pr-041812

[10] Kumar, S., Jacob, V.S. and Sriskandarajah, C. "Scheduling Advertisements On A Web Page To Maximize Revenue," *European Journal of Operational Research* 173(1), 2006, 1067-1089.

[11] Kumar, S., Dawande, M. and Mookerjee, V. "Optimal Scheduling and Placement of Internet Banner Advertisements," *IEEE Transactions on Knowledge and Data Engineering*, 19(11), 2007, 1571-1584.


## Appendix A — GA Parameter and Setting Definitions

**GA:**

Population Size (ps) — the number of chromosomes (solution strings) which are included in each population

Selection Process — process by which parent chromosomes are selected from the population

Mutation Process — process by which individual bits are to be selected as potential mutation candidates

Mutation Probability ( $p_m$ ) — the probability that a mutation candidate is mutated

Crossover Process — process by which parental chromosomes are combined to form the children chromosomes

Stopping Criteria — criteria which determines how many generations are included in the GA run

Coding Scheme — process by which potential solutions are coded are coded as chromosome strings

Elite List Percentage ( $e$ ) — the percentage of chromosomes which survive from one generation to the next in their current form.