Mathematical models for the television advertising allocation problem

Xinhui Zhang

Department of Biomedical, Industrial and
Human Factors Engineering, Wright State University,
207 Russ Engineering Center, 3640 Colonel Glenn Hwy,
Dayton, OH 45435, USA
Fax: (937) 775-7364 E-mail: xinhui.zhang@wright.edu

Abstract: Television networks deliver television programming to the public free of charge; their primary source of revenue is the sale of advertising slots in their programmes. A key problem faced by the TV networks is how to allocate these slots to advertisers. The problem is complicated by sophisticated show structure, limited inventory of slots, demographics, show preferences and competition avoidance.

In this paper, a two step hierarchical approach is proposed to solve this problem. This approach starts with a winner determination problem to select advertisers and assign them to shows and ends with a pod assignment problem to schedule commercials of the selected advertisers in a show. The winner determination problem is solved using column generation algorithm which was able to get near optimal solutions orders of magnitudes faster than the state-of-the-art B&B algorithm. The methodology proposed offers great potential for these networks to increase revenue.

Keywords: column generation; hierarchical approach; integer programme; optimisation; television advertising; winner determination.

Reference to this paper should be made as follows: Zhang, X. (2006) 'Mathematical models for the television advertising allocation problem', *Int. J. Operational Research*, Vol. 1, No. 3, pp.302–322.

Biographical notes: Xinhui Zhang is an assistant professor of industrial engineering at Wright State University. He received his PhD in Operations Research and Industrial Engineering from the University of Texas at Austin in 2003. Dr. Zhang's research interests include mathematical programming and optimisation in the fields of manufacturing, transportation, media management, service operations and engineering design. He has participated and led several projects such as, the airline crew recovery project, the advertising allocation for television networks, the equipment and staff scheduling for mail processing facilities and the design and optimisation of the VLSI test system. Dr. Zhang's publications have appeared in *IIE Transaction*, *Computer and Operations Research* and *European Journal of Operations Research*.

1 Introduction

Television network companies, such as ABC, NBC and CBS, and their affiliated stations, are in the business of delivering television programming to the public free of charge. In doing so, television networks attract a large variety of viewers and, therefore, draw much attention from advertisers – broadcasters intend to attract audiences to 'sell' them, and advertisers desire to 'buy' them. Of the various television viewing periods, prime time attracts the greatest number of viewers and receives the most attention from both the networks and the advertisers.

Television prime time includes all regularly scheduled programmes from 8:00 p.m. to 11:00 p.m. (EST), Monday through Saturday, and 7:00 p.m. to 11:00 p.m. on Sunday. Due to its large viewing audience, advertisers are willing to pay large sums of money to place a commercial in these shows. For example, for the 2000–2001 season, Table 1 reports the average cost to air a 30-second advertisement on the three major networks ABC, NBC and CBS, while Table 2 shows the cost to air a 30-second advertisement for some top-rated shows on NBC (Mandese, 2001). As seen in Table 1, a 30-second advertisement on NBC costs an average of $198,864. The price on a top-rated show during prime time, as shown in Table 2, is much higher: $620,000 during 'ER', $540,000 during 'Friends', and $465,000 during 'Just Shoot Me'.

Table 1 Average cost for a 30-second advertisement for the 2000–2001 season

ABC	*NBC*	*CBS*
$193,409	$198,864	$190, 114

Table 2 Cost of a 30-second advertisement during top-rated shows for the 2000–2001 season

ER	*Friends*	*Just Shoot Me*
$620,000	$540,000	$465,000

These figures clearly demonstrate the importance of advertising to television networks. Unlike other media companies such as cable networks which get part of their revenue from subscriptions, television networks rely heavily on the advertising income. One of the key problems faced by these networks is how to allocate the commercial slots in their shows to advertisers so as to maximise revenue. In this paper, this problem is referred as the advertising allocation problem.

Solving this advertising allocation problem is not an easy task. The problem is complicated by sophisticated show structure, limited inventory of advertising slots, audience demographics, show preference, and competition avoidance to maintain the effectiveness of the advertisement. The solution to this problem calls for the use of optimisation techniques. While optimisation has been successfully employed in various other fields, such as the airline industries (Yu, 1997), manufacturing and service industries (Zhang and Bard, 2005), only a few studies exist that address the optimisation problems in the television and media industry.

The majority of these studies, however, focus on scheduling television programmes, rather than commercials, to maximise certain criteria such as audience size. The problem here is how to rely on the strength of the preceding programme to boost the rating of a newly introduced programme. For some of these models, see Horen (1980), Reddy et al. (1998), Rust (1986) and Rust and Echambadi (1989).

Only a handful of these studies addressed the allocation of commercials to advertisers in an effort to maximise the networks' revenue. Bollapragada and Garbiras (2004) and Bollapragada et al. (2002) studied the commercial scheduling problem to generate sales plans to meet the requirement of a single advertiser. The problem was modelled as an integer programme and solved sequentially for each advertiser with the objective of making the least use of premium inventory. Bollapragada et al. (2004) then studied the problem on scheduling commercials over a specific period (for example, a month) so that the airing of the same commercials are spread as evenly as possible. Several integer programmes were proposed and heuristics were developed to solve the problem efficiently.

Jones (2000) introduced the advertising allocation problem as an example to design incompletely specified combinatorial auctions where potentially hundreds of advertisers can submit combinatorial bids for the airing of their commercials in the advertising slots. The problem was modelled as an integer programme that took into consideration detailed locations of the time slots. The model was computationally prohibitive and heuristics based on constraint programming were used to find feasible solutions.

The research proposed in this paper is based on the advertising allocation scenario proposed in Jones (2000). However, rather than building an all-inclusive model that is impossible to solve, a more manageable approach is to develop models that strike a balance between accuracy and computational tractability. To this end, a two-step hierarchical approach is proposed. Firstly, a winner determination problem is solved to select advertisers and assign them to shows. Then, a pod assignment problem is used to schedule selected advertisers' commercials to slots within a specific show.

This two-step hierarchical approach calls for the solution of two optimisation programmes. While the pod assignment problem, a quadratic integer programme, is trivial to solve, the winner determination problem, a large scale general integer programme, has posed a great challenge even to the state-of-the-art branch and bound (B&B) algorithm. To tackle this problem, a column generation algorithm is developed. This algorithm is composed iteratively solving a set-packing type master problem and a set-covering type subproblem. Computation results demonstrated the superiority of this customised algorithm, which was able to obtain near optimal solutions orders of magnitude faster than B&B algorithm. For large problems, while B&B was unable to obtain a feasible solution within 12 hours of computation, the column generation algorithm was able to get solutions within 97% optimal in only 20 minutes. These models and solution approaches are applicable to similar advertising allocation problems in radio and cable networks.

The remainder of the paper is organised as follows. Section 2 describes the prime time advertising allocation problem and proposes the two-step hierarchical approach. Section 3 presents the winner determination problem and its formulation. The details of the column generation algorithm are presented in

Section 4 and computation results are shown in Section 5. Section 6 discusses the pod assignment problem and concluding remarks are given in Section 7.

2 The television advertising allocation problem

Television networks sell most of their advertising slots in what is called the upfront market. This market opens right after the networks have established their schedules for the next season and sells approximately 60–80% of the commercial slots inventory. The research focuses on the sale of the advertising slots in this market.

2.1 Problem description

The problem addressed here is how should television networks select advertisers and allocate advertising slots to them to maximise their revenue? The allocation has to consider various factors, such as show structure, audience demographics, show preference, unduplicated viewers and competition avoidance.

2.1.1 Show structure

Show structure refers to the length of a show and the number of commercial breaks during the show. For instance, an hour-long show typically contains five to seven commercial breaks; each break is referred to as a pod. Moreover, each pod may last one to two minutes and contains a set number of advertising slots; the exact number is dependent on the exact length of the pod. For this study, one advertising slot is 15 seconds in length and is referred to as one unit.

2.1.2 Audience demographics

Audience demographics refer to audience characteristics, such as age, gender, income, education and lifestyle. These characteristics receive high attention from both networks and advertisers who are primarily interested in reaching viewers that are potential customers of their products. In television advertising, gender and age are considered the most important. Typically, six demographic categories are associated with prime time television advertising: women, 18 to 49; women, 25 to 54; men, 18 to 49; men, 25 to 54; adults, 18 to 49; and adults, 25 to 54. Audience size within these demographics categories measures the relative success of a show. However, without some sort of direct feedback from each viewer, it is impossible for television networks to document the exact audience size of a show. As a result, networks and advertisers resort to rating services, such as Nielsen, that use sampling techniques to obtain estimates of viewership.

2.1.3 Show preference

Besides age and gender, other demographic information, such as income, education and lifestyle, may also be of value to the advertiser, but this information is typically not provided by the rating services. To better reach these specific subsets of audience groups, advertisers will specify a set of preferred shows during which to advertise their commercials. In this paper, it is assumed that the advertisers will stipulate a set of preferred shows during which to advertise and a minimum percentage of allocated units in preferred shows.

2.1.4 Unduplicated viewers

Unduplicated viewers refer to the viewers who tuned in a programme at least once for a few minutes. A common phenomenon being observed in the industry is that the same people will show up in the audience several times. While this frequent recurrence is good for making a commercial more effective, it reduces the number of unduplicated audience members exposed to the commercial. To achieve maximum exposure to an unduplicated audience, advertisers usually set an upper bound on the maximal number of commercials allowed in a single show. This allows them to spread their commercials to as many shows and, hopefully, to as many different individuals as possible. Setting this upper bound for a show to zero, on the other hand, helps advertisers to disallow any allocation in an undesired show.

2.1.5 Competition avoidance

To avoid competition between advertisements, television networks have to ensure that an allocation schedule complies with the following rules:

- pod anti-competition, which stipulates that competing advertisements do not appear in the same pod
- anti-cluttering, which is imposed to prevent the allocation of an overwhelming large number of different commercials in a single pod.

Both cluttering and competition in a pod dilute the strength of the advertiser messages and are to be avoided if possible.

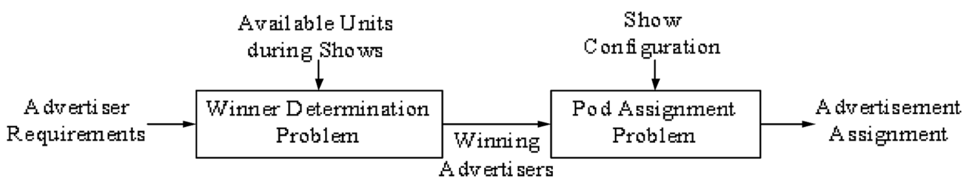
A network sells a set of programmes or shows, each of which may last as short as 30 minutes or as long as three hours. The network maintains a reserved price on one unit of commercial time in each show and desires to sell these slots to advertisers to maximise the revenue while satisfying the advertisers' demographic coverage and show preference requirements. An advertiser may have a composite campaign with commercials of different lengths to promote several products and is willing to pay a premium if his requirements are met.

2.2 A two-step hierarchical approach

Ideally, it would be desirable if the problem could be solved with an all-inclusive model, but this is not possible due to the overwhelming complexity of the requirements. Rather, our experiences in other industries (Zhang and Bard, 2005) have shown that a hierarchical approach usually works best.

The major decisions in the sale of commercial time slots are the selection of advertisers (winners) and the assignment of their commercials to shows. This fact is also observed in Jones (2000) where the author stated that "many of the decisions necessary to discover an optimal allocation of goods can be determined at an aggregate or show level rather than at the unit or the pod level." In view of this, a two-step hierarchical approach is developed. This approach consists of solving two optimisation problems, a winner determination and a pod assignment problem sequentially and is depicted in Figure 1.

Figure 1 A two-step hierarchical approach for the advertising allocation problem



This hierarchical approach starts with the winner determination problem, which select advertisers and assign their commercials to shows. This problem takes as inputs, the advertisers' demographic requirements and the show inventory and determines winning advertisers with an objective of maximising revenue. In an attempt to achieve a balance between model accuracy and computation tractability, the winner determination problem is modelled on show level, rather than on pod level, and does not take into account the anti-competition requirements.

Given the winning advertisers and their guaranteed allocation of slots during each show, a pod assignment problem is then solved to schedule advertisers to pods. The objective here is to satisfy as many competition avoidance requirements as possible through minimum adjustment to the structure of a show, such as the number of units in a pod or the number of pods in a show.

3 The winner determination problem

3.1 Mathematical model

The winner determination problem is a simplified advertising allocation problem without consideration of the competition avoidance requirements. It is the first step and the centre of the hierarchical approach. To fully describe the model, the following notation is used.

Indices

b	Index of advertisers
s	Index of television programmes or shows
g	Index of demographic groups
n	Index of commercials

Sets

B	Set of advertisers
S	Set of television programmes or shows
G	Set of demographic groups
$S(b)$	Set of preferred shows of advertiser b
$N(b)$	Set of commercials of advertiser b

Parameters

v Total number of units available in all shows

u_s Number of available units in show s

c_s Reserved price for a unit in show s

θ_{sg} Number of viewers (in Nielsen ratings points) of group g exposed to one unit in show s

p_b Price for the overall demographic requirements of advertiser b

d_{bg} Gross coverage requirement (in Nielsen rating point) in category g of advertiser b

l_n Length (in unit) of commercial n, $n \in N(b)$

m_{bsn} Maximum number of times commercial n of advertiser b is allowed in show s, $n \in N(b)$

ρ_b Ratio of the number of units allocated in preferred shows to that in all shows

Decision variables

x_{bs} Number of units assigned to advertiser b in show s

y_{bsn} Number of times commercial n of advertiser b appeared in show s

z_b Binary variable representing whether or not an advertiser b's bid is accepted

The mathematical model can be formulated as follows:

$$\text{Maximise } \sum_{b \in B} p_b z_b - \sum_{b \in B} \sum_{s \in S} c_s x_{bs} \tag{1}$$

subject to:

$$\sum_{s \in S} \sum_{b \in B} x_{bs} \le v \tag{2}$$

$$\sum_{b \in B} x_{bs} \le u_s \qquad \forall s \in S \tag{3}$$

$$x_{bs} \le u_s z_b \qquad \forall b \in B,\ s \in S \tag{4}$$

$$\sum_{s \in S} \theta_{sg} x_{bs} \ge d_{bg} z_b \qquad \forall b \in B,\ g \in G \tag{5}$$

$$\sum_{s \in S(b)} x_{bs} \ge \rho_b \sum_{s \in S} x_{bs} \qquad \forall b \in B \tag{6}$$

$$x_{bs} = \sum_{n \in N(b)} l_n y_{bsn} \qquad \forall b \in B,\ s \in S \tag{7}$$

$$y_{bsn} \le m_{bsn} \qquad \forall b \in B,\ s \in S,\ n \in N(b) \tag{8}$$

$$x_{bs} \ge 0,\ y_{bsn} \ge 0,\ \text{and integer} \qquad \forall b \in B,\ s \in S,\ n \in N(b) \tag{9}$$

$$z_b \in \{0, 1\} \qquad \forall b \in B. \tag{10}$$

The first term of the objective function represents the network's advertising revenue from winning advertisers while the second term represents the reserved price of the programmes. The difference between the two terms provides the network's net revenue which is to be maximised.

Constraint (2) ensures that the total number of units allocated in all the shows cannot exceed the total number of units available for sale. Notice that constraint (2) is redundant if $v \geq \sum_{s \in S} u_s$; however, for the upfront market, only 60–80% of the advertising slots are sold. Constraint (3) guarantees that the number of units allocated will not exceed the inventory in a show.

Constraint (4) enforces that if an advertiser's request is not accepted ($z_b = 0$), then no airtime is allocated; while constraint (5) guarantees that if an advertiser's request is accepted ($z_b = 1$), then all his demographic requirements have to be satisfied. Constraint (6) represents the preferred show requirements: the summation of units allocated in preferred shows must exceed a certain percentage, ρ_b, of the summation of units allocated during all shows. Constraint (7) guarantees that an assignment must be a valid combination of different commercial lengths. Constraint (8) keeps track of the upper bound on the number of commercial units. Finally, constraints (9) and (10) state that all decision variables are integer-valued.

3.2 Model analysis

The above model contains $2 \times |B| \times |S| \times |\bar{N}|$ integer variables and almost an equal number of constraints. Here, $|B|$ represents the number of advertisers, $|S|$, the number of shows, and $|\bar{N}|$ the average number of commercials of an advertiser. For an upfront market, a typical problem contains 30–50 shows and attracts as many as 200–300 advertisers, resulting in a large-scale integer programme. As an example, a problem consisting of 300 advertisers and 48 shows contains 32, 172 integer variables and 29,449 constraints, and presents a great challenge to even the best branch and bound (B&B) algorithm. However, the problem clearly exhibits a block angular structure and can be tackled with a Dantzig-Wolfe decomposition algorithm discussed below.

4 A Dantzig-Wolfe decomposition/column generation algorithm

4.1 Background on Dantzig-Wolfe decomposition

Dantzig-Wolfe decomposition is a well-known technique and has been successfully applied in solving large-scale integer programmes such as the airline crew scheduling problem and the vehicle routing problem. For a general survey of this technique, see Barnhart et al. (1998). One way to apply this decomposition scheme to the winner determination problem is to include constraints (2)–(3) to create the main problem and to leave constraints (4)–(10) to form the subproblem. The main problem incorporates the show capacity constraints, while the subproblem defines the advertiser's demographic requirements. The main problem has few constraints, would converge quickly, and the subproblem could be solved efficiently either through optimisation or through heuristics.

4.2 Dantzig-Wolfe decomposition reformulation

To apply the above Dantzig-Wolfe decomposition scheme, the problem must be reformulated. To do so, let $X_b = \left\{x_b^1, x_b^2, \cdots, x_b^{K_b}\right\}$ be the set of feasible solutions to advertiser b, i.e. a solution $x_b^k = \left\{x_{b1}^k, x_{b2}^k, \cdots x_{b,|S|}^k, z_b^k\right\}$ satisfies constraints (4)–(10). Superscript k represents a different solution. In addition, notice that when $z_b^k = 0$, all $x_{b1}^k, x_{b2}^k, \cdots, x_{b,|S|}^k$ should be 0, as such, only solutions where $z_b^k = 1$ need to be considered. Furthermore, let λ_b^k, $k \in \{1, 2, \ldots, K_b\}$, be a binary variable representing whether or not such a non-zero feasible solution x_b^k is selected. The original model, model (1)–(10), can then be reformulated as

$$\text{Maximise:} \sum_{b\in B}\sum_{k=1}^{K_b} p_b\lambda_b^k - \sum_{b\in B}\sum_{k=1}^{K_b}\lambda_b^k\sum_{s\in S}c_s x_{bs}^k = \sum_{b\in B}\sum_{k=1}^{K_b}\left(p_b - \sum_{s\in S}c_s x_{bs}^k\right)\lambda_b^k \tag{11}$$

subject to:

$$\sum_{b\in B}\sum_{s\in S}\sum_{k=1}^{K_b} x_{bs}^k\lambda_b^k \leq v \tag{12}$$

$$\sum_{b\in B}\sum_{k=1}^{K_b} x_{bs}^k\lambda_b^k \leq u_s, \qquad \forall s \in S \tag{13}$$

$$\sum_{k=1}^{K_b}\lambda_b^k \leq 1 \qquad \forall b \in B \tag{14}$$

$$\lambda_b^k \in \{0, 1\} \qquad \forall k = 1, \cdots K_b,\ b \in B. \tag{15}$$

The objective function represents the network's net revenue. Constraints (12) and (13) state that units assigned to all advertisers cannot exceed the total advertising inventory and the capacity of each show respectively. Constraint (14) states that out of an advertiser's feasible assignments, at most one can be chosen. Constraint (15) stipulates that λ_b^k must be an integer.

Model (11)–(15), referred to as the reformulated model, is an integer programme with a large number of variables. To solve it optimally, a branch and bound algorithm has to be developed, but first the solution of the linear programme (LP) relaxation of this model needs to be examined.

4.3 The linear programme relaxation of the reformulated model

The LP relaxation of the reformulated model is obtained by replacing, in (15), the integrity requirement on λ_b^k with a non-negative requirement only; however, even this LP relaxation of the reformulated model cannot be handled directly due to the large number of variables.

To resolve this difficulty, a restricted problem, where only a subset of variables is present, is first solved. Given this subset of columns, the LP relaxation can be solved with, say, a simplex method. Let $\alpha \geq 0$, $\beta_s \geq 0$, and $\gamma_b \geq 0$ be the dual variables associated with constraints (12), (13) and (14) respectively. Now, checking whether the optimal solution to the LP relaxation of the reformulated model is achieved or not depends on whether there exists any column λ_b^k, $k = 1, \cdots K_b, b \in B$, with a positive reduce cost, which is represented as

$$p_b - \sum_{s \in S} c_s x_{bs} - \alpha \sum_{s \in S} x_{bs} - \sum_{s \in S} \beta_s x_{bs} - \gamma_b$$

Thus, optimality checking can be done by solving the following problem.

$$\text{Maximise: } p_b - \sum_{s \in S} (c_s + \alpha + \beta_s) x_{bs} - \gamma_b \tag{16}$$

subject to: constraints (4)–(10).

If the solution is greater than 0, then a column with positive reduced cost exists. The column is then added to the restricted master problem and the process is repeated. Otherwise, no column with positive reduced cost exists and the LP relaxation of the master problem is solved optimally. Notice that the subproblem decomposes by advertisers and thus can be solved efficiently. The above algorithm generates additional columns to the master problem and therefore is also called column generation. The reformulated model with all the columns included is referred as the master problem and the one with only a subset of the columns included as the restricted master problem (RMP). The optimality checking problem is referred to as the pricing subproblem.

4.4 Branch and bound and branching rules

Upon solving LP relaxation of the master problem, if all λ_b^k, $k = 1, \cdots K_b, b \in B$ are integer, then the optimal integer solution to the original model is obtained. Otherwise, a fractional solution occurs. To get integer solutions, the above algorithm has to be embedded into a branch and bound procedure and branching must be performed. The conventional way of branching on fractional λ_b^k, however, is not effective because it destroys the structure of the pricing subproblem: though it is easy to set λ_b^k to 1 and force a column to be in the solution, it is difficult to set λ_b^k to 0 and keep a column out in the solution of the subproblem.

The key to designing branching rule is to branch on variable in the original formulation. This means that branching should be performed on the original variable x_{bs}. To do so, it must be shown that for a fractional solution to reformulated model, there is an advertiser b and a show s such that x_{bs} is factional. This is true because otherwise all $x_{bs} = \sum_{k=1}^{K_b} \lambda_b^k x_{bs}^k$ are integer and an integer feasible solution to the master problem is obtained. As such, suppose in the LP solution, $\sum_{k=1}^{K_b} \lambda_b^k x_{bs}^k = \varphi$ is a fraction, then two branches $x_{bs} \leq \lfloor \varphi \rfloor$ and $x_{bs} \geq \lceil \varphi \rceil$ can be created. This branch rule can be easily implemented in the subproblem by setting upper and lower bounds on x_{bx} without destroying the structure of the subproblem.

4.5 Approximation algorithm

An alternative way of getting a high quality solution to the original problem is to solve the reformulated model with only a limited number of columns included. This can be done easily as follows. Firstly, apply the above column generation scheme to solve the LP relaxation of reformulated model to price out columns. Then the integrity requirement on λ_b^k is brought back and the problem is solved as an integer programme with all the columns available. The complete column generation approximation procedure is presented below.

- *Step 1*: (Initialisation) Set all dual variables α, β_s, and γ_b to 0 and solve the corresponding pricing subproblem for each advertiser to generate initial columns for the RMP.
- *Step 2*: (Main problem) Solve the RMP as a linear programme and find the dual costs α, β_s, and γ_b associated with each constraint in the master problem.
- *Step 3*: (Pricing subproblem) Solve the pricing problem for each advertiser. If no solutions with objective greater than zero exist, go to Step 4. Otherwise, add the solutions with positive objective values as columns to the RMP and go to Step 2.
- *Step 4*: (Optimality check) If all the variables in the RMP are integer, the original problem is solved and an optimal solution is obtained; go to step 6. Otherwise, the LP solution is fractional; go to Step 5.
- *Step 5*: (Approximation) Solve the RMP with all the columns generated so far as a 0–1 integer programme and find the optimal solution.
- *Step 6*: Stop.

Because not all columns are presented when reformulated model in step 5 is solved, the solution obtained is feasible but not necessarily optimal to the original problem. However, as will be seen in the next section, the approximation algorithm was able to yield a high quality solution in an acceptable amount of time. Due to the extensive amount of time required to solve the whole problem optimally, a full column generation algorithm embedded in a branch and bound procedure is not used in the final implementation.

5 Computational results

5.1 Data generation

To fully describe the advertising allocation problem, two sets of data are necessary:

- show configuration, such as the number of units in a show, show's performance, the reserved price, and the total number of shows
- advertiser's request, such as demographic coverage, the length of commercials, the set of preferred shows, the preference weight, the price and the total number of advertisers.

However, industry information, particularly pricing and show inventory availability, is proprietary and heavily protected, and thus are not readily available.

The basis of our data generation was established on the statistics of two weeks of airtime allocation provided by a major television network in Jones (2000). Nevertheless, additional assumptions, such as the correlations of a) an advertiser's price and his demographic requests and b) an advertiser's preference of a show and the show's performance, have to be made. Effort has made to mimic the advertising sale as close as possible.

Depending on the number of advertisers and the number of shows, the problems are classified into three groups: small-size problems, where $|B| = 50$ and $|S| = 8$; medium-size problems, where $|B| = 100$ and $|S| = 16$; and large-size problems, where $|B| = 200$ and $|S| = 32$. It was also observed that the more demographic categories a show can cover, the harder the allocation problem becomes. By varying the average number of demographic categories a show covers, denoted as $|\bar{G}_s|$, the problems can be further classified within each group as easy ($|\bar{G}_s| = 1.5$), hard ($|\bar{G}_s| = 2$), and hardest ($|\bar{G}_s| = 2.5$).

5.2 Computational results

A comprehensive set of experiments was conducted to evaluate the quality of solutions produced by the proposed column generation approximation algorithm to reformulated model as well as to compare the solution to the standard B&B algorithm applied to the original model. All models were implemented using *Mosel*, the modelling language provided by Dash Optimization Inc. and solved using *Xpress*, their general linear, integer and quadratic programme solver. All computations were performed on a 1.13 GHZ Pentium III PC with 512 MB of RAM. Both the pricing subproblem and the approximation integer programme were solved using *Xpress*. Though these problems can be solved more efficiently using heuristics, the use of *Xpress* is meant to simply demonstrate the effectiveness of the proposed column generation algorithm.

5.2.1 Comparison of upper bounds

Both the original model and the reformulated model are equivalent integer programmes but differ in their linear programming relaxations. For a maximisation problem, both of their LP relaxations provide upper bounds for the integer programme. However, it is well known that because the pricing subproblem is not totally unimodular, the LP relaxation of the reformulated model, denoted as Z^{CG}, is a better bound than that the LP relaxation of original model, denoted as Z^{BB}. Our first experiment is designed to answer how much better the solution Z^{CG} could be.

Table 3 presents the upper bounds provided by these two relaxations. In the table, the '$|B|$' column represents the number of advertisers, '$|S|$', number of shows and '$|\bar{G}_s|$' the average number of demographic categories a show covers. For comparison purposes, the best bound obtained for the original model when the B&B algorithm has reached its time limit is also reported. These values are listed under the 'Z^{BBT}' column and the time limits are set at 12 hours for large-size problems and two hours for medium-size problems. 'Gap 1' refers to the gap between Z^{CG} and Z^{BB} and is calculated as $(Z^{BB} - Z^{CG})/Z^{CG}$. 'Gap 2' refers to the gap between Z^{CG} and Z^{BBT} and is calculated as $(Z^{BBT} - Z^{CG})/Z^{CG}$.

As shown in Table 3, Z^{CG}, the LP bound of reformulated model was the best of the three bounds. On average, Z^{BB}, the LP bound of the original model is 6.1% higher than Z^{CG}. For a large problem, this gap is even higher, 8.1%. Z^{CG} is still better than Z^{BBT}, the best bound obtained after hours of computation. For large problems, Z^{BBT} on average is 3.6% higher than the Z^{CG}.

Table 3 Comparison of upper bounds (Z^{CG} and Z^{BB})

Cases			*Bounds*				
$\|B\|$	$\|S\|$	$\|\bar{G}_s\|$	Z^{BB}	Z^{BBT}	Z^{CG}	*Gap 1 (%)*	*Gap 2 (%)*
200	32	2.5	1,761,770	1,711,893	1,663,098	5.9	2.9
200	32	2	1,754,104	1,704,064	1,652,029	6.2	3.1
200	32	1.5	1,469,317	1,372,392	1,309,684	12.2	4.8
100	16	2.5	813,728	772,737	753,006	8.1	2.6
100	16	2	737,739	685,031	675,729	9.2	1.4
100	16	1.5	427,438	415,371	415,920	2.8	−0.1
50	8	2.5	279,588	259,308	259,433	7.8	−0.1
50	8	2	211,042	191,435	207,757	1.3	−7.8
50	8	1.5	128,114	121,595	126,420	1.3	−3.8

5.2.2 Solution quality

Our second experiment is designed to evaluate and compare the quality of solutions obtained when the approximation algorithm is applied to solve the reformulated model with that when the branch and bound algorithm is applied to solve the original model.

Table 4 presents the quality of solutions associated with the approximation and the B&B algorithm for the two models. Feasible solutions obtained from each algorithm, if available, are listed in 'Obj (IP)' column. The 'Gap' column under column generation refers to the gap between the feasible solution and the bound obtained from the approximation algorithm. The 'Gap' under B&B refers to the gap between the feasible solution and the best bound obtained when the time limits were reached. The 'Time root' column under the approximation algorithm denotes the time to complete the root node. The 'Time IP' column under approximation algorithm represents the time limits to solve resulting 0–1 integer programme. The time limits to solve the B&B algorithm are set as before at 12 hours for large-size problems and two hours for medium-size problems. The times reported for small-size problems are the time used to reach the optimal solution. The bounds and objective values of the column generation approximation algorithm, and the times used in these two algorithms, are presented. The bound under column generation algorithm refers to the objective value obtained while solving the root node.

As shown in Table 4, small-size problems were trivial to solve. Both the B&B algorithm and the approximation algorithm were able to obtain the optimal solutions quickly. B&B algorithm was also able to prove that the solutions were optimal; however, as can be seen, when the problem size increased, the performance of B&B algorithm deteriorated dramatically. For medium-size problems, the gap between the feasible solution and the best bound obtained after two hour computation was 3.4% in the 100-16-1.5 case, 10.8% for the 100-16-2 cases, and 15.8% in the 100-16-2.5 case respectively. No feasible solution was ever found for any of the large-size problems when the 12-hour time limit was reached.

Table 4 Comparison results of the winner determination problem

Scenario			*Column generation*					*Branch and bound*			
$\lvert B\rvert$	$\lvert S\rvert$	$\lvert\bar{G}_s\rvert$	*Bound (root LP)*	*Time root*	*Obj (IP)*	*Time IP*	*Gap %*	*Best bound*	*Obj (IP)*	*Gap %*	*Time limit*
200	32	2.5	1,663,098	1147	1,612,948	60	3.0	1,711,893	*	*	12 h
200	32	2	1,652,029	466	1,616,477	60	2.1	1,704,064	*	*	12 h
200	32	1.5	1,309,684	57	1,268,668	60	3.1	1,372,392	*	*	12 h
100	16	2.5	753,006	37	742,693	10	1.4	772,737	667,408	15.8	2 h
100	16	2	675,729	22	650,572	10	3.7	685,031	617,625	10.8	2 h
100	16	1.5	415,920	19	402,302	10	3.3	415,371	401,423	3.4	2 h
50	8	2.5	259,433	6	259,308	2	0.1	259,308	259,308	0.0	6 s
50	8	2	207,757	3	191,435	2	7.8	191,435	191,435	0.0	2 s
50	8	1.5	126,420	2	121,595	2	3.8	121,595	121,595	0.0	1 s

Note: *No feasible solution obtained when the time limit was reached

The approximation algorithm, on the other hand, consistently provided high quality optimal or near optimal solutions. For the medium-size problems, the solutions obtained for the three problems were within 96.7, 96.6 and 98.7% optimal, respectively. These solutions were 11.3, 5.3 and 0.2% better than the corresponding solutions from the B&B algorithm. More importantly, these solutions were obtained within one minute. Compared to two hours for the B&B algorithm, the approximation algorithm is more than two orders of magnitude faster.

For large-size problems, the algorithm was able to get solutions that were within 98.3, 97.9 and 96.9% optimal in 20 minutes, eight minutes and one minute, respectively. Recall that the B&B algorithm was not able to obtain any feasible solution after 12 hours of computation for any of these problems. In fact, even if the time limit was lifted, the B&B algorithm did not provide any integer solution and was finally stopped due to running out of memory after more than a day of computation. The reason behind this poor performance, we believe, is primarily due to the inferior LP relaxation bound of the original formulation, which leads to a large number of infeasible nodes being explored and slows the overall convergence of the algorithm.

Except in a few trivial cases, the approximation algorithm provided a better bound and yielded near-optimal solutions two to three orders of magnitudes faster than the B&B algorithm. To further verify the accuracy of the column generation algorithm, we seeded the solutions to the middle-size problems into the B&B algorithm. The solutions were promptly confirmed feasible, but no improved solution was obtained after another hour of computation.

5.2.3 The convergence of the column generation process

Our third experiment is designed to examine the convergence speed of the approximation algorithm and the quality of solution for extreme large-size problems when the algorithm has to be terminated prematurely due to the limited amount of computation time available. In this experiment, all the columns obtained during each iteration of the column generation algorithm were included to form the reformulated integer programme. The programme was then solved and the solution was evaluated.

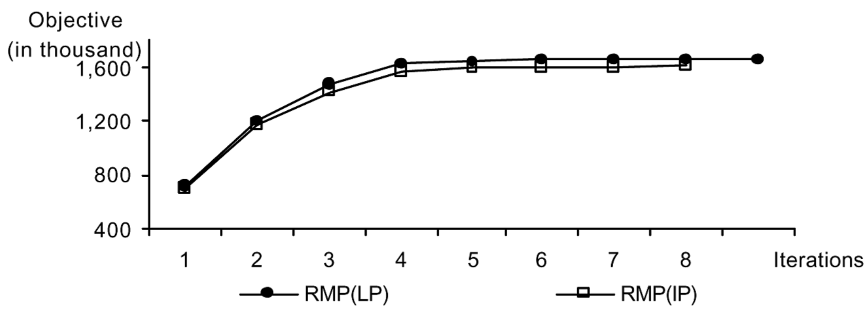
Table 5 presents the column generation process for the 200-32-2.5 allocation problem. For this problem, a total of nine iterations were required to solve the LP relaxation of the master problem and they are indexed below the 'Iteration' row. The 'RMP (LP)' row records the objective value of the linear relaxation of each RMP and the 'RMP(IP)' row shows the integer solution obtained of each integer RMP programme. The 'Gap' row measures the gap between the integer solutions of the nth RMP, the value under the corresponding 'RMP(IP)' row, and the LP relaxation of the final master problem. The 'Increase' row keeps track of the percentage increase of the integer solution of the nth RMP from the previous one. Finally, the 'Time (sec)' row refers to the accumulated time in seconds to reach to nth iteration. The time limit to solve the resulting integer programme was set at 60 seconds.

Table 5 The convergence speed and solution quality of the column generation process

Iterations	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
RMP(LP)	708,140	1,202,721	1,466,940	1,630,476	1,649,089	1,659,661	1,662,793	1,663,098	1,663,098
RMP(IP)	695,035	1,167,499	1,412,105	1,564,942	1,595,491	1,595,741	1,599,878	1,612,948	1,612,948
Gap	41.8%	70.2%	84.9%	94.1%	96.0%	96.0%	96.2%	97.0%	97.0%
Increase	–	69.84%	21.97%	11.15%	1.95%	0.64%	0.18%	0.02%	0.00%
Time (sec)	184 s	204 s	218 s	231 s	274 s	453 s	634 s	826 s	1147 s

From Table 5, it is apparent that good solutions can be obtained at the early stages of the column generation process. Though the solution from the iteration 1 was only 41.8% optimal, two iterations later, iteration 3, the solution was 84.9% optimal; the ratio almost doubled. At iteration 5, the solution was 96.0% optimal, only 1% away from the final integer solution. The qualities of solutions improved slowly afterwards and are shown in Figure 2.

Figure 2 The column generation process (the 200-32-2.5 allocation problem)



The slow improvement at the end of a column generation process is known as the tail effect and is generally not preferred because of the long computation time required to achieve an optimal solution. However, the tail effect is beneficial because it implies that terminating the column generation process prematurely due to limited computation time would not significantly undermine the quality of the solution obtained. For example, if the column generation process is terminated when the increase of objective is less than 1%, from Table 5, it would stop at Iteration 6. At this stage, the whole process took only 453 seconds. Even with the addition of another 60 seconds to solve the resulting integer programme, the whole process would have taken only 513 seconds, i.e. less than ten minutes. Compared with the B&B approach where no solution is obtained after 12 hours, column generation would have already found a solution within 96% of optimal in less than ten minutes. That is nearly three orders of magnitude faster.

5.2.4 The solution of RMP

To close this section, it has to be mentioned that finding the solution of the resulting integer programme in Step 5 of the approximation algorithm is not always an easy task. For the 200-32-2.5 allocation problem, the RMP integer programme for the 9th iteration contains 233 rows and 1041 columns and is rather difficult to solve. When the RMP integer programme was solved using *Xpress*, a solution with an objective value of 1,612,948, or 97% optimal, was obtained when a time limit of 60 seconds was reached. This is the solution reported in Table 5. However, when the time limit was raised, better solutions could be obtained. Solutions with values of 1,630,135, and 1,634,637 emerged at 366 seconds and 863 seconds respectively. The final solution was 98.4% optimal, which again confirms the superiority of the column generation algorithm. Because the upper bound from the LP relaxation is often higher than the optimal solution, this final solution could be much closer to optimal than indicated.

The effective solution of the integer RMP can further improve the performance of the overall approximation algorithm and demands more attention. This problem is a combinatorial optimisation problem and may be solved efficiently using neighbourhood search heuristics, such as simulated annealing and tabu search.

6 The pod assignment problem

The pod assignment problem is invoked after the winner determination problem is solved. The winner determination problem focuses on the assignment of advertisers to shows and thus does not take into account the competition avoidance requirements. Recall that to avoid diluting the strength of each commercial, it is desirable not to allocate competing commercials in the same pod. However, as mentioned, adding these pod-level requirements into the winner determination model requires significantly more variables and constraints and the model would be too large to solve for any problem of realistic size.

In practice, television networks have considerable flexibility in revising the configuration of the show, such as modifying the number of units during commercial breaks and perhaps even the number of commercial breaks before it is actually aired. The pod assignment problem utilises these flexibilities to schedule commercials to pods and to meet as much of the competition avoidance requirements as possible through minimum adjustment to a show.

This pod assignment problem decomposes by show, allowing one show to be addressed at a time. As such, index s is dropped from the notation used in the development of the model.

Indices

p	Index of pods in the show
c	Index of groups of competitors
n	Index of commercials
t	Index of competition types

Sets

P	Set of pods in a show
B	Set of advertisers assigned to the show, solution from the winner determination problem
T	Sets of competitor types
$N(b)$	Set of commercials of advertiser b
$B(t)$	Set of advertisers of competition type t

Parameters

u_p	Number of units in pod p
$\bar{y}_{bn}$	Number of times commercial n of advertiser b is allocated in the show, solution of y_{bsn} from the winner determination problem, but index s is dropped
l_n	Length (in unit) of commercial n, $n \in N(b)$
θ_1, θ_2	Weights on the two objective terms

Decision variables

x_p Actual number of units assigned in pod p

y_{bpn} Number of times commercial n of advertiser b is allocated in pod p

z_{bp} Binary variable representing whether or not advertiser b is allocated in pod p

w_{pc} Number of competitions in a pod p of type competition t

The mathematical model can be formulated as follows:

$$\text{Minimise: } \theta_1 \sum_{p \in P} (x_p - u_p)^2 + \theta_2 \sum_{t \in T} \sum_{p \in P} w_{pt}^2 \tag{17}$$

subject to

$$\sum_{p \in P} y_{bpn} = \bar{y}_{bn} \qquad \forall n \in N(b),\ b \in B \tag{18}$$

$$\sum_{b \in B} \sum_{n \in N(b)} l_n y_{bpn} = x_p \qquad \forall p \in P \tag{19}$$

$$y_{bpn} \leq u_p z_{bp} \qquad \forall p \in P,\ n \in N(b),\ b \in B \tag{20}$$

$$\sum_{n \in N(b)} y_{bpn} \geq z_{bp} \qquad \forall p \in P,\ b \in B \tag{21}$$

$$\sum_{b \in B(t)} z_{bp} \leq w_{pt} + 1 \qquad \forall t \in T,\ p \in P \tag{22}$$

$$y_{bpn} \geq 0 \text{ and integer} \qquad \forall p \in P,\ n \in N(b),\ b \in B \tag{23}$$

$$z_{bp} \in \{0, 1\} \qquad \forall p \in P,\ b \in B. \tag{24}$$

The objective here is to minimise a weighted sum of two quadratic terms, the first one being the number of changes in unit to pods and the second one being the number of competitions in the show. Constraint (18) guarantees that an advertiser's committed allocation for each commercial obtained from the winner determination problem is satisfied. Constraint (19) keeps track of the total number of units assigned to a pod. Constraints (20) and (21) define whether an advertiser is allocated in a pod or not. Constraint (22) records the number of competitions in a pod p. Finally, constraints (23) and (24) stipulate that all variables are integer.

The pod assignment problem is small in size because the problem decomposes by show and only winning advertisers are involved in the solution of each show. Consequently, though the pod assignment problem is a quadratic integer programme, it can be solved quickly, usually in seconds. Computational results show that generally no modification to the show structure was necessary unless the number of competing advertisers in a group exceeded the number of pods in a show. In the cases where competition cannot be avoided, violations were evenly distributed among pods and were minimal. The reason seems to be that not all slots are sold in the upfront market, thus allowing extra slots to be used to move competing advertisers in a show.

7 Summary

In this paper, the television advertising allocation problem is studied and a two-step hierarchical framework is presented. This framework, composed of a winner determination problem and a pod assignment problem, captures the essence of advertising assignment to shows while offering computationally tractability. The winner determination problem is the kernel of this framework and a column generation algorithm is developed to solve it efficiently. The algorithm was able to get near optimal solutions orders of magnitudes faster than the state-of-the-art B&B algorithm, and is critical to the efficiency of the whole framework. The models and solution approaches proposed are applicable to advertising allocation problems in cable networks and radio stations where less restrictive anti-competition requirements are imposed. An efficient allocation of advertising slots to advertisers will be able to give these networks great potential to increase revenue and profitability.

There are two tasks that need to be addressed to extend the model in the advertising market. Firstly, the research is based on the analysis of two typical weeks of advertisement requirements and assumes that buyers will purchase airtime continuously from week to week. In reality, continuous campaigns are rare. Advertisers want to place an advertisement in specific weeks actively to suit their needs. These active weeks are routinely followed by periods of inactivity or reduction in the exposure requirements. To model time windows to accept buyer specific airdates while still offering computational tractability needs further consideration. Secondly, the research adopts an auction environment where all buyers' requests are known beforehand so the selection of advertisers is essentially a deterministic optimisation problem. In reality, the television advertising sale is still in a negotiation process where advertiser requests come sequentially. Though television networks have certain knowledge of the advertisers' requests from past sales, these requests do change constantly and thus introduce uncertainty. How to accept or reject advertising requirements in the negotiation process under uncertainty is another task to be addressed.

Acknowledgements

This research is supported by an Ohio Board of Regents Research Challenge Award granted by Wright State University. This research is an extension of the study the author conducted for Net Exchange, the company that designs and builds combined value markets.

References

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P. and Vance, P.H. (1998) 'Branch and price: column generation for solving huge integer problems', *Operations Research*, Vol. 46, No. 3, pp.316–329.

Bollapragada, S. and Garbiras, M. (2004) 'Scheduling commercials on broadcast television', *Operations Research*, Vol. 52, No. 3, pp.337–345.

Bollapragada, S., Bussieck, M. and Mallik, S. (2004) 'Scheduling commercial videotape on broadcast television', *Operations Research*, Vol. 52, No. 5, pp.679–689.

Bollapragada, S., Cheng, H., Philips, M., Garbiras, M., Scholes, M., Gibbs, T. and Humhreville, M. (2002) 'NBC's optimization systems increase revenues and productivity', *Interfaces*, Vol. 32, No. 1, pp.47–60.

Horen, J.H. (1980) 'Scheduling of network television programs', *Management Science*, Vol. 26, No. 4, pp.354–370.

Jones, J.J. (2000) 'Incompletely specified combinatorial auction: an alternative allocation mechanism for business to business negotiations', PhD Dissertation, University of Florida, FL.

Mandese, J. (2001) ' "ER" tops price charts, regis wears the crown', *Advertising Age*, Vol. 71, pp.1–2.

Reddy, S.K., Aronson, J.E. and Stam, A. (1998) 'SPOT: scheduling programs optimally for television', *Operations Research*, Vol. 44, No. 1, pp.83–102.

Rust, R.T. (1986) *Advertising Media Models: A Practical Guide*, Lexington, MA: Lexington Books.

Rust, R.T. and Echambadi, N.V. (1989) 'Scheduling network television programs: a heuristic audience flow approach to maximize audience share', *Journal of Advertising*, Vol. 18, pp.11–18.

Yu, G. (1997) *Operations Research in the Airline Industry*, Boston, MA: Kluwer Academic Publishers.

Zhang, X. and Bard, J.F. (2005) 'Equipment scheduling at mail processing and distribution centers', *IIE Transactions on Scheduling & Logistics*, Vol. 37, No. 2, pp.175–187.