텔레비전 광고 스케줄링 문제의 최적화 해법

An Optimization Algorithm for the Advertisement Scheduling Problem on Broadcast Television

이지원·박성수
한국과학기술원 산업공학과

Abstract

We consider the advertisement scheduling problem on broadcast television in a given period. Major advertisers want to air several kinds of commercials and purchase broadcast time slots for them without considering actual scheduling. When same commercials are aired multiple times, they usually need to be aired as regularly as possible. This problem is called the basic Industry Standard Commercial Identification (ISCI) rotator problem. The objective of the problem is to assign commercials to be aired as regularly as possible. We propose an integer programming formulation of the problem. Contrary to the previous model, the new model has many columns, hence we solved it by branch-and-price(B&P) approach and computational results are reported. We tested our algorithm on 40 instances that have various slot size.

1. Introduction

Major advertisers purchase hundreds of time slots to air commercials during any broadcast season without considering actual scheduling. During the broadcast season, each company decides the actual scheduling for several commercials and sends videotapes and instruction. Each videotape includes one type of commercial and it has a unique code that is called to Industry Standard Commercial Identification (ISCI) code. A broadcasting station should arrange them to purchased slots by the advertiser well according to advertisers' instructions about the number of times each commercial has to be aired during the broadcasting season. In the case, advertisers usually want air their commercials as regularly as possible in a given period. Hence, this problem is how to arrange same kind of commercials of an advertiser to be distributed evenly while satisfying the advertiser's instruction.

This problem is called to the basic ISCI rotator problem or, the ISCI problem, simply. Example of this problem is described in table 1 and table 2.

Table 1. Time slot purchased by an advertiser

Program	Day	Air time
Sports news	Mon (10/25)	18:10 PM
Happy time	Tue (10/26)	17:50 PM
Ani world	Wed (10/27)	16:50 PM
Sports news	Wed (10/27)	18:10 PM
19:00 news	Thu (10/28)	19:00 PM
19:00 news	Fri (10/29)	19:00 PM

Table 2. Commercial type and the number of times to be aired

ISCI Code	Frequency
ABCD1234	2
IPOS2453	1
XLWK1694	1
QLLV3461	2

When the type of color and the number of all balls of the color corresponds to the type of commercial and the number of same type of commercials to be aired respectively, this problem can be stated formally as follows: We have N balls and the number of types of ball color is C. The number of all balls of each color i is n_i. We have N available slots and assign each ball to a slot. We should try to assign balls of same colors as evenly spaced as possible. When the distance of subsequent balls of color i is as close as possible to N/n_i, we can say that the balls are distributed evenly as much as possible. This problem was presented by Bollapragada et al. (2002) at first and Bollapragada et al. (2004) presented a formulation and heuristic algorithms for the problem. In this paper we propose an integer programming formulation of the problem and solve it by branch-and-price (B&P) algorithm.

The remainder of this paper is organized as follows. In Section 2 we define notation and present a set partitioning model of the ISCI problem. Section 3 describes a branch-and-price algorithm for the ISCI problem. Computational results are reported in Section4, and Section5 contains some conclusions.

2. Model

We consider several color types and assume that the number of all balls is given for each color. If we can find all possible configurations for each color, the ISCI problem can be a problem that is to decide which configuration can be used for each color. The following notation is used:

Π : the set of colors

Ω : the set of slots

C : the number of color types

n_i : the number of balls of color $i(\in \Pi)$

N : total number of balls, $N = \sum_{i\in\Pi} n_i$ (it is equals total number of slots)

q_i :ideal distance of color $i(\in \Pi)$, $q_i = N/n_i$

Q_i : the set of all configurations for color $i(\in \Pi)$

c_i^q : cost to assign balls by configuration $q(\in Q_i)$ of color $i(\in \Pi)$

$$f_{ij}^q = \begin{cases} 1, & \text{: if a configuration } q(\in Q_i) \text{ of color } i(\in \Pi) \text{ uses slot } j(\in \Omega), \\ 0, & \text{: otherwise.} \end{cases}$$

A configuration for $q(\in Q_i)$ of color i($\in \Pi$) can be represented by $f_i^q = (f_{i1}^q, .., f_{i,N}^q) \neq 0$. The configuration f_i^q means a set of slots that are selected for the color. We calculate cost of the configuration f_i^q with the distance of balls that are assigned by the configuration, that is, the sum of deviation of the distance of subsequent balls of color i to the ideal distance of the color. Ideal distance is a measure how well balls of same color are distributed evenly among all. A slot can be assigned to one ball. The decision variable is as follows:

$$\lambda_i^q = \begin{cases} 1, & \text{: if a configuration } q(\in Q_i) \text{ of color } i(\in \Pi) \text{ is chosen,} \\ 0, & \text{: otherwise.} \end{cases}$$

We construct a formulation for the master problem:

[MP]

$$\text{Min} \quad \sum_{i\in\Pi}\sum_{q\in Q_i} c_i^q \lambda_i^q \quad (1)$$

$$\text{s.t.} \quad \sum_{i\in\Pi}\sum_{q\in Q_i} f_{ij}^q \lambda_i^q = 1, \quad \forall j \in \Omega \quad (2)$$

$$\sum_{q\in Q_i} \lambda_i^q = 1 \quad , \quad \forall i \in \Pi \quad (3)$$

$$\lambda_i^q \in \{0,1\}, \quad \forall i \in \Pi, q \in Q_i \quad (4)$$

The objective that is represented by (1) is to minimize total cost of selected configurations. Constraint (2) imposes that each slot can be assigned to one ball. A configuration f_i^q corresponds to a column vector $(f_{i1}^q, f_{i2}^q, ..., f_{iN}^q)'$ of this constraint. Each color can choose one configuration by constraint (3).

Bollarpragada et al. (2004) presented a flow formulation for this ISCI problem. Corresponding network uses N nodes and a dummy node 0 that is an artificial source and sink. Arcs from each node j (including node 0) to all node k ($j < k \le N$) are existed and all node j (not including node 0) have a directed arc whose tail node is j and head node is 0. Two binary decision variables for the network are considered. The first decision variable is an assignment variable p_{ij} whose value is one if color i uses node j. The second one is a flow variable f_{ijk} whose value is one if ball of color i is shipped from node j to node k. A ball of color i is shipped at node 0 and passes through n_i nodes and goes back to node 0. c_{ijk} is corresponding cost coefficient of flow variable f_{ijk} and it is calculated as $|k - j - q_i|$. We call the flow formulation as $M1$ and define Z_S as the objective value of the LP relaxation of formulation S.

Proposition1. $Z_{MP} \ge Z_{M1}$

Proof. Let $\delta = (\delta_1^1, ..., \delta_i^k, ..., \delta_C^{|Q_C|})$ be a feasible solution set of MP. Then the objective value of MP is $\xi = \sum_{i\in\Pi}\sum_{q\in Q_i} c_i^q \delta_i^q$. Configuration f_i^q of $q(\in Q_i)$ th column of color $i(\in \Pi)$ can be used to assign value of elements of an assignment vector $p_i^* = (p_{i1}^*, ..., p_{iN}^*)$ of M1 for color $i(\in \Pi)$ as follows:

$$p_{ij}^* = \sum_{q\in Q_i} f_{ij}^q \delta_i^q, \quad \forall i \in \Pi, \ j \in \Omega$$

Also, we define a flow vector $x_i^* = (x_{i01}^*, x_{i12}^*, ..., x_{i,N-1,N}^*, x_{i,N,0}^*)$ for color $i(\in \Pi)$ of M1. Initial value of all elements of x_i^* is zero. Then every configuration f_i^q of color $i(\in \Pi)$ such that $\delta_i^q > 0$ can be used to assign value of elements of flow vector x_i^* for each color $i(\in \Pi)$ as follows. We add flow value δ_i^q to x_{ijk}^* for all pairs $j(\in \Omega)$ and $k(\in \Omega)$ such that $f_{ij}^q = 1 \ and \ f_{ik}^q = 1$ and $f_{it}^q = 0$ for all $t = j+1, ..., k-1$. Also we add flow value δ_i^q to x_{i0l}^* for $l(\in \Omega)$ such that $f_{il}^q = 1$ and $f_{it}^q = 0$, for all $t = 1, ..., l-1$. Similarly we add same flow value to $x_{i,l',0}^*$ for $l'(\in \Omega)$ such that $f_{il'}^q = 1$ and $f_{ik}^q = 0$ for all $k = l'+1, ..., N$. Then p_i^* and x_i^* is feasible to M1 and $\sum_{q\in Q_i} c_i^q \delta_i^q = \sum_{0<j<k\le N} c_{ijk} x_{ijk}^*$ is satisfied for all color $i(\in \Pi)$.

For every feasible solution set δ of MP, we can construct feasible vector p_i^* and x_i^* of M1 for each color $i(\in \Pi)$ and the objective value of M1 is $\sum_{i\in\Pi}\sum_{0<j<k\le N} c_{ijk} x_{ijk}^* = \sum_{i\in\Pi}\sum_{q\in Q_i} c_i^q \delta_i^q = \xi$. It is same with the objective value of MP. It completes the proof.

□

3. Algorithm

We developed a branch-and-price algorithm for the ISCI problem. To enumerate all columns is not effective because there are too many columns and most columns are not used in MP, therefore we take subset of all possible columns. The master problem with the subset of all columns is called to restricted master problem. We optimize the LP relaxation of the restricted master problem. We can generate configurations that can be used in the restricted master problem by solving subproblem. The objective of subproblem is to find a configuration that has minimum reduced cost. If the reduced cost of the configuration is negative, the configuration will be entered into the restricted master problem as a column. We repeat this procedure until there is no configuration that has negative reduced cost. If there is no configuration that has negative reduced cost and the optimal solution of the LP relaxation of the restricted master problem is not integral, branching occurs. By two-phase method, we can ensure a feasible solution of the LP relaxation exists at every branching node. Since the master problem is set partitioning model, generating an artificial variable is enough. See Barnhart et al. (1998).

3.1 Subproblem

The subproblem can be solved for each color. We can transform the subproblem into the shortest path problem easily by constructing a network. Necessary notation is as follows:

π_j : dual variable for constraint (2), $j \in \Omega$

μ_i : dual variable for constraints (3), $i \in \Pi$

s,t : an artificial source and sink node, respectively

c_{ijk} : cost of directed arc whose head node is k and tail node is j, such that $j < k$, $j,k \in \Omega$ and $i \in \Pi$, $c_{ijk} = | k - j - q_i |$

For the subproblem of color $i (\in \Pi)$, we generate n_i layers that each layer includes N nodes and add downward arcs between every subsequent layer. Downward arc is an arc whose node number of tail node is less than node number of head node. We also add an artificial source node s and an artificial sink node t. With node s, we construct directed arcs whose tail node is node s and whose head node is every node in layer 1. Similarly, we construct directed arcs whose tail node is every node in layer n_i and whose head node is node t. Figure 1 represents the network that we describe. Note that every s-t path has same path length.

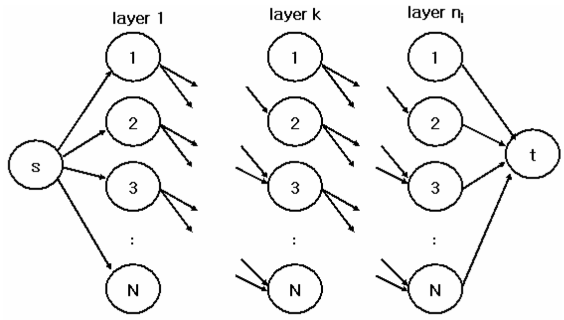

Figure 1. A network for subproblem of color i

We assign arc cost of this network as follows. Cost of every arc that is incident to node s is zero and cost of directed arcs that exist between every subsequent layers is $c_{ijk} - \pi_j$ if its head and tail node is node $k (\in \Omega)$ and node $j (\in \Omega, j < k)$, respectively. Cost of every arc that is incident to node t is $-\pi_j$ when its tail node is node $j (\in \Omega)$. For every layer, some subset of nodes will never be chosen when we generate any feasible s-t path because only downward arcs are allowed. By not allowing the use of arcs incident to the nodes, we can save time to search for paths. For any $s-t$ path of the network, a node that can be chosen in layer l is one of subset of slot $j (\in \Omega)$ that satisfies $l \le j \le N - n_i + l$. Cost of arcs that is incident to the node that is not included the subset of nodes that can be chosen in feasible $s-t$ path of layer l will be fixed to M, where M is large positive number.

By solving the $s-t$ shortest path of the network, we can construct a new configuration $f_i^{q'}$ for color $i(\in \Pi)$. For all $j(\in \Omega)$ th element of configuration $f_i^{q'}$, value of $f_{ij}^{q'}$ is one if node j is included in the $s-t$ shortest path, otherwise value of $f_{ij}^{q'}$ is zero. Cost of the $s-t$ shortest path for color $i(\in \Pi)$ is $c_i^{q'} - \sum_{j \in N} f_{ij}^{q'} \pi_j$, where $f_i^{q'}$ is a corresponding configuration to the path and $c_i^{q'}$ is cost to assign all balls of color $i(\in \Pi)$ by configuration $f_i^{q'}$. Namely, we can get a minimum reduced cost of color $i(\in \Pi)$ by subtracting dual value μ of color $i(\in \Pi)$ from cost of the $s-t$ shortest path, where reduced cost of color $i(\in \Pi)$ is $c_i^{q'} - \sum_{j \in N} f_{ij}^{q'} \pi_j - \mu_i$. The new configuration $f_i^{q'}$ is added to the subset of configurations Q_i' of color $i(\in \Pi)$ if corresponding reduced cost is negative. Since this network is an acyclic digraph, we can get a $s-t$ shortest path in time $O(m)$, where m is the number of arcs. See Cook et al. (1998).

3.2 Branching strategy

If an optimal solution λ of the restricted master problem is fractional, branching occurs. We introduce $p_{ij} = \sum_{q \in Q_i} f_{ij}^q \lambda_i^q$ for $j \in \Omega$, $i \in \Pi$. If color $i(\in \Pi)$ uses slot $j(\in \Omega)$, $p_{ij} = 1$. Otherwise, $p_{ij} = 0$. When there is a fractional λ_i^{q*}, Then there should exist a color $i(\in \Pi)$ and a partitioning row $j(\in \Omega)$ of constraint (2) such that $0 < p_{ij} < 1$, hence the pair of branching constraints is given by $p_{ij} = 1$ (left branch) and $p_{ij} = 0$ (right branch). We choose the pair that has most fractional value p_{ij} and when ties, we select to branch on slot j of color i such that $\pi_j (\sum_{q=1}^{|Q_i|} f_{ij}^q \lambda_i^q - 1)$ is largest (See Fisher (1981)). After branching, we keep generating columns and we can easily get a configuration by adjusting the network that is constructed for the subproblem. Also when we optimize the LP relaxation of the master problem after branching, some columns should not be used any more. It can be resolved easily by adjusting upper bound of some columns. Let us define the branching color and branching slot as i^* and j^*, respectively.

At the left branch, we modify every network for each color as follows. For the network of branching color i^*, we should use node j^* exactly once. Therefore at the first layer, we should choose a node whose node number is less than or equal to j^*. At the last layer we choose a node whose node number is greater than or equal to j^*. And at all layers, nodes whose node number are less than j^* can not have any arc that is incident to nodes whose node number is greater than j^*. For the networks of other color $k(\in \Omega, \neq i^*)$, we should not use node j^* at all layers. We can not use all incoming arcs to node j^* and outgoing arcs from node j^*. Cost of forbidden arcs of networks is fixed to M, where M is very large positive number. Every detail is described at Figure 2 and 3. Figure 2 is for color i^* and Figure 3 is for other color $k(\in \Omega, \neq i^*)$. Dotted lines in Figure 2 and 3 means that they can not be used any more after branching. In the master problem, some columns can not be used after branching. Among columns for color i^*, columns that do not use slot j^* will not be used. Among columns for others, columns that use slot j^* are not also considered. We can make the columns not used any more by fixing upper bound of corresponding variables to zero.

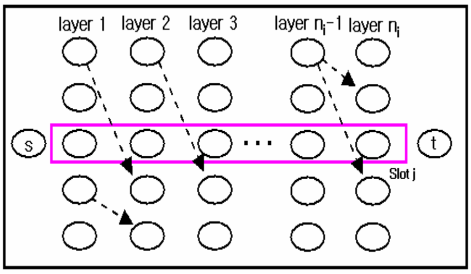

Figure 2. Modified Network (1)

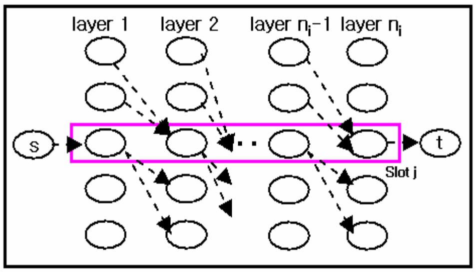

Figure 3. Modified Network (2)

At the right branch, we only modify a network of subprobem for branching color i^*. The network should not use node j^* at all layers. Use of incoming arcs to node j^* and outgoing arcs from node j^* of all layers is forbidden. It is well described in Figure 3. Columns of color i^* of the master problem are not used if the value of j^* th element of the column is one. We fix cost of forbidden arcs of the network and upper bound of columns of the master problem to M and zero, respectively.

To control the tailing-off effect, we compared the best possible lower bound $\underline{z}^u_{LP}$ with the objective value $z^u(\bar{Q})$ of the LP relaxation of the restricted master problem at the node u (See Vanderbeck and Wolsey (1996)). We consider the least common multiple LCM of the number of balls of every color. If $\left\lceil \underline{z}^u_{LP} \cdot LCM \right\rceil \geq z^u(\bar{Q}) \cdot LCM$, column generation at the node is terminated. If the number of balls of different color is same, their subproblem generate same configuration and it causes undesirable degeneracy (See Lubbecke and Desrosiers (2002)). Therefore we make them share column pool each other.

4. Computational results

We coded the procedures in C language and used callable library of CPLEX 9.0. All problems were tested on a Pentium Ⅳ (2.8GHz). We used test problem set presented by Bollapragada et al. (2004). It contains forty instances whose slot size is distributed from 8 to 400. We divided them into three classes. The first one is “small size” problem that includes problem 1~10. The second part is consisted of problem 11~22 and this is said to “usual size” problem of broadcast television industry. Last part that is a set of problem 23~40 is considered “large”. All problems are consisted of at least two kinds of color and at most five kinds of color. All process was terminated if the execution time is over two hours. Computational results are given in the Table 3.

Table 3. The test results (small size)

#	N	Z_{IP}	Z_{LP}	GAP	Cols	B_B	Time
1	8	2.67	2.67	0.00	13	0	0.0
2	8	0.00	0.00	0.00	9	0	0.0
3	10	2.33	0.833	0.64	46	6	0.0
4	11	3.00	2.50	0.16	57	5	0.0
5	12	3.00	3.00	0.00	45	0	0.0
6	12	2.00	0.00	1.00	45	3	0.0
7	14	4.67	4.67	0.00	68	0	0.0
8	15	4.25	3.25	0.25	20	3	0.0
9	16	3.33	1.67	0.50	147	11	1
10	17	5.39	4.44	0.18	130	4	0

Table 3. The test results cont. (usual size)

#	N	Z_{IP}	Z_{LP}	GAP	Cols	B_B	Time
11	20	5.79	4.36	0.25	170	6	0
12	20	7.17	5.02	0.30	105	9	4
13	20	3.00	3.00	0.00	25	0	1
14	25	6.79	3.98	0.41	128	8	8
15	25	8.46	4.90	0.42	259	8	73
16	30	9.25	3.46	0.62	231	4	114
17	30	10.1	5.40	0.47	720	11	56
18	40	13.7	6.82	0.50	1446	14	223
19	45	6.22	6.22	0.00	15	0	0.0
20	50	3.68	3.68	0.00	13	0	0.0
21	50	19.5	13.0	0.33	2516	19	1723
22	50	14.5	4.81	0.67	2358	19	7000

Table 3. The test results cont. (large size)

#	N	Z_{IP}	Z_{LP}	GAP	Cols	B_B	Time
23	60	22.8	20.3	0.18	6159	15	5129
24	75	33.8	15.6	0.54	6212	43	3659
25	100	62.7	32.9	0.48	7250	-	7200
26	100	38.5	29.1	0.24	6135	37	2795
27	150	22.0	22.0	0.00	1500	0	3150
28	200	90.1	55.9	0.38	785	-	879
29	200	150.7	63.3	0.58	895	-	5450
30	250	165.6	65.26	0.60	1564	-	7200
31	300	179.9	139.8	0.22	1400	-	7200
32	300	137.5	44.7	0.67	1352	-	7200
33	325	155.6	98.5	0.37	1245	-	7200
34	350	326.2	154.4	0.53	1456	-	7200
35	400	230.7	-	-	-	-	7200
36	400	215.0	154.2	0.28	2564		7200
37	425	277.2	-	-	-	-	7200
38	450	390.6	-	-	-	-	7200
39	500	472.2	-	-	-	-	7200
40	500	375.4	-	-	-	-	7200

In the table, the heading “#” and “N” means instance number and the slot size of the problem set respectively. Z_{IP} means the best feasible solution of the ISCI problem and Z_{LP} is the optimal value of the LP relaxation of the MP. The value Z_{IP} without underline represent it is an optimal solution for the instance and underlined Z_{IP} represent it is a best feasible solution in two hours. GAP is calculated as $(Z_{IP} - Z_{LP}) / Z_{IP}$. The heading “Cols” and “B&B” means the number of columns and the number of branching that were generated to get Z_{IP} of the instance, respectively. Time(sec) is the computational time to get an optimal solution of the problem. All small size problems were solved in short time. Most problems whose size is usual were solved in five minutes. For large size problem, we could get an optimal solution of four instances in two hours.

5. Conclusions

In this paper, we present an optimization algorithm for the ISCI problem using branch-and-price algorithm. By the algorithm, we can get an optimal solution for the problems that are small or typical size in broadcast television in a short time. For the large size problems, time to solve was long. It is because speed to solve the LP relaxation of the master problem is too slow at an iteration and subproblem has no restriction other than the number of balls of each color. To speed up, alternative formulation of the problem is possible.

As a further research, we consider problems that reflect realistic constraints. In real situation, unit length of each advertisement can be different and some commercials prefer specific slots.

REFERENCES

Barnhart, C., E.L. Johnson, G.L. Nemhauser and M.W.P.Savelsbergh (1998), “Branch and price: Column generation for solving huge integer programs”, *Operations Research* 46(3), 316-329.

Bollapragada, S., Hong Cheng, Mary Phillips and Marc Garbiras, Michael Scholes, Tim Gibbs, Mark Humphreville (2002), “NBC’s optimization systems increase revenues and productivity”, *Interfaces* 32(1), 47-60.

Bollapragada, S., Michael R. Bussieck, and Suman Mallik (2004), “Scheduling commercial videotapes in broadcast television”, *Operations Research* 52(5), 679-689.

Lubbecke, Macro E., and Desrosiers, J. (2002), “Selected topics in column generation”, to appear in Operations Research

Marshall L. Fisher. (1981), “The lagrangian relaxation method for solving integer programming problems”, *Management Science* 27(1), 1-18

Vanderbeck, F., Wolsey, L.A., (1996), “An exact algorithm for IP column generation”, *Operations Research Letters* 19, 151-159.

William J. Cook, William H. Cunningham, William R. Pulleyblank, Alexander Schrijver, (1998), *Combinatorial Optimization*, Wiley-Interscience series in discrete mathematics and optimization.