



Research article

A branch and bound algorithm for optimal television commercial scheduling

Lu-Wen Liao*

Department of Intelligent Production Engineering, National Taichung University of Science and Technology, Taichung 40401, Taiwan

* **Correspondence:** Email: lwliao@gm.nutc.edu.tw.

Abstract: In the current era of multimedia, television (TV) plays an important role in transmitting advertising messages. In addition, advertising is the primary source of revenue for the TV industry. Thus, a critical issue for TV stations is the scheduling of commercials in suitable advertising breaks on different TV channels to maximize revenue and minimize penalties. This type of TV commercial scheduling problem is similar to the machine scheduling problem, and both have availability constraints. However, the literature on TV commercial scheduling has not considered this perspective. Motivated by this, we consider the problem of scheduling commercials with specific service-level requirements on TV channels while minimizing the maximum lateness. The lateness of a commercial is defined to be its completion time minus its due date, and the maximum lateness is the maximum value of lateness among all commercials. We propose an exact branch and bound algorithm based on the *LFJ* (*least flexible job first*)/*EDD* (*earliest due date first*) rules and network flow methods, which were developed to solve the machine scheduling problem with availability constraints. Computational analysis shows that the bounding scheme proposed is highly effective, and a very low percentage of nodes is generated by the branch and bound algorithm. The algorithm can obtain an optimal solution for the problem.

Keywords: TV commercials; service level requirement; scheduling; *LFJ/EDD* rule; branch and bound algorithm

1. Introduction

Television (TV) plays an important role in transmitting advertising messages in the current era of multimedia. In fact, TV advertising is one of the most effective channels to provide fast propagation

of advertising messages to audiences [1]. In India, TV is the largest and fastest growing medium and is expected to grow at a compound annual growth rate of 15%, with total revenue reaching USD 15.2 billion in 2019 [2]. According to an NBC Universal report, its TV advertising business generated more than 5 billion U.S. dollars in revenue each year from 2012 to 2020 [3]. The website Statista predicted that television advertising revenue in the United States will grow from 72.3 billion U.S. dollars in 2021 to 81 billion in 2025. Further, global TV advertising revenue has been projected to increase from 151 billion U.S. dollars to 162 billion between 2020 and 2024 [4]. Thus, TV advertising, which is the primary source of revenue for the TV industry, is very big business [5] and [6].

In TV advertising, there are two major participants: TV channels (or networks/stations) and advertisers. TV channels need to schedule programs into appropriate time slots or spots (advertising breaks) for the next scheduling horizon. Then, the marketing department of the TV channel sells the spots based on associated audience ratings. Advertisers buy the advertising breaks with the goal of achieving the most efficient media plan within a limited budget [7]. Thus, TV advertising scheduling allocates a number of advertisement time slots to advertisers while satisfying advertisers' requirements or choosing the advertisers to whom the time slots are sold [8]. This is one of the most important decisions for a TV channel, providing a vital source of revenue for the TV station [9]. This type of scheduling is becoming more and more competitive and turbulent because of the fast development of communication and broadcasting means. Over the past two decades, this challenging problem has attracted much research.

Brown [10] discussed various difficulties in scheduling TV commercials manually and developed an algorithm to exchange commercials of different lengths between advertising slots to gain space for other commercials. Hägele, Ó Dúnlain, and Riis [11] proved that the problem defined by Brown [10] is NP-complete. Bollapragada et al. [12], Bollapragadae et al. [13] and Brusco [14] dealt with the advertising scheduling challenges faced by NBC. Bollapragada et al. [12] defined an optimization model to generate sales plans to meet the requirements of a single advertiser. The problem was modelled as mixed-integer programming, and solutions were found sequentially for each advertiser using mathematical programming-based heuristics.

Bollapragada et al. [13] and Brusco [14] also discussed the scheduling of multiple commercials such that the output schedule allows for the appropriate spacing of multiple airings of the same commercials. Brusco [14] developed an enhanced version of the branch and bound algorithm of Bollapragada et al. [13] and a simulated annealing heuristic for handling larger instances of the problem studied. Bollapragada and Garbiras [15] formulated the commercial scheduling problem as a goal programming model and designed a two-stage heuristic algorithm for the automatic scheduling of commercials that minimized the penalty cost for the television network while incorporating the costs of sales personnel and meeting promised positions with consideration of product conflicts. Gaur, Krishnamurti, and Kohli [16] extended the model proposed by Bollapragada and Garbiras [15]. They considered the differential weighting of conflicts between the pairs of commercials by formulating the problem as a binary integer program that was a generalization of the max k-cut.

Ghassemi-Tari and Alaei [5] considered the concept of a combinatorial auction and formulated a combinatorial auction-based mathematical programming model to maximize the revenue of the television network via commercial placement. García-Villoria and Salhi [7] considered the possibility of multiple airings of each commercial being as evenly spaced as possible during the airing period. Furthermore, advertisers may request a minimum number of times for which each commercial is aired in high- and medium-audience-rating spots. Thus, García-Villoria and Salhi [7] presented two mixed-

integer linear programming models, local search procedures, and a simulated annealing algorithm to solve the proposed problem.

Guerriero, Miglionico, and Olivito [17] addressed the problem of accepting commercials for TV slots with the aim of maximizing TV revenue. They developed mathematical programming models and defined several constructive heuristics together with some local searches to solve the problem. Díaz-Núñez, Halman, and Vásquez [9] discussed a TV channel considering a TV rating points function and willingness to pay coefficients with the aim of scheduling a set of TV commercials in a single time slot to maximize the total revenue. Sinha, Dharmarha, and Kumar [1] proposed a goal programming model to determine the optimal schedule of commercials for a TV channel, advertising breaks, and the frequency of commercials.

Czerniachowska [6] considered the scheduling problem from the perspective of the advertisers, proposing a genetic algorithm to allocate TV commercials according to the requirements of the advertisers and their limited budget while maximizing the total viewership. Fontes, Pereira, and Fontes [8] considered the problem of scheduling self-promotion commercials to a given set of advertising breaks while maximizing audience ratings between TV programs for a TV station. For comprehensive surveys in this area, we refer the readers to [2].

Considering the literature described above, we find that a critical issue faced by TV channels is how to schedule commercials in advertising breaks to maximize the revenue or minimize the penalties incurred for not satisfying the requirements of the clients. This is similar to the machine scheduling problem, which also faces availability constraints. The machine availability constraint denotes a set of identical machines that are not continuously available for processing; in other words, the machines can only serve jobs on a set of availability intervals [18,19]. To the best of our knowledge, the existing literature has not considered the TV advertising scheduling problem from this perspective. Thus, we are motivated to deal with the TV advertising scheduling problem by applying the methods developed for solving the machine scheduling problem with availability constraints. Therefore, we propose a branch and bound algorithm to address the TV commercial scheduling problem.

The remainder of the paper is organized as follows. In Section 2, we describe the problem and notations used throughout this paper. In Section 3, we construct a branch and bound algorithm for the problem. Computational experiments are given in Section 4, and Section 5 concludes.

2. Definition of the problem and notations

In a typical parallel machine scheduling problem with machine availability constraints, there are n jobs, each with given processing time and due date, and m machines, which are only available for processing in specified availability intervals. The goal is to find a schedule that minimizes the maximum lateness. The lateness of a job is defined to be its completion time minus its due date. The maximum lateness problem consists of finding a schedule that minimizes the maximum lateness over all jobs. In other words, jobs should be completed at available times on machines as close as possible to their due dates. In this paper, the studied problem is similar to the stated parallel machine scheduling problem with machine availability constraints.

Next, we define the scheduling in the context of TV advertising as follows. We consider the problem of scheduling n commercials with service-level requirements on m TV channels. Each commercial has a due date, release time, processing time, and required service level. Each TV channel has a number of advertising breaks. These advertising breaks are classified into different levels

according to certain properties, such as the audience rating on TV programs, which is used as the service level in this study. The service level constraint denotes that a commercial can only be processed within a specified set of the advertising breaks. A lower service-level requirement of a commercial can be satisfied by advertising breaks with the same or higher service level. However, a TV channel that provides a lower service level cannot broadcast commercials that require higher service levels. The objective is to minimize the maximum lateness of the schedule, as most TV channels do not want to delay the time taken to broadcast their clients' commercials because the delay can incur penalties.

Table 1. Notations.

| Set and indices | |
|---------------------|--|
| A | Set of TV channels, indexed by i |
| B | Set of commercials, indexed by j |
| E | Set of service levels, indexed by k . The service level is highest when $k = 1$ and lowest when $k = h$ |
| F | Set of the advertising break, indexed by r |
| M_j | Set of TV channels that can process commercial j , $j \in B$ |
| R_k | Set of commercials requiring service level k , $k \in E$ |
| IS_k | Set of advertising breaks with service level k , $k \in E$ |
| SL_j | $SL_j = \bigcup_{l=1}^k IS_l$; the set of advertising breaks that can serve commercial j with service level k , $j \in B$ and $k \in E$ |
| J | Set of commercials j already scheduled, $j \in B$ |
| J^c | Set of commercials j not yet considered for scheduling, $j \in B$ |
| Parameters | |
| h | Total number of service levels |
| r_j | Release time of commercial j , $j \in B$ |
| p_j | Processing time of commercial j , $j \in B$ |
| rs_j | Required service level of commercial j , $j \in B$ |
| d_j | Due date of commercial j , $j \in B$ |
| C_j | Completion time of commercial j , $j \in B$ |
| $N(i)$ | Total number of advertising breaks on TV channel i , $i \in A$ |
| $[b_i^r, f_i^r]$ | r th advertising break on TV channel i , $i \in A$ and $r \in F$ |
| b_i^r | Start time of the r th advertising break on TV channel i , $i \in A$ and $r \in F$ |
| f_i^r | End time of the r th advertising break on TV channel i , $i \in A$ and $r \in F$ |
| rs_i^r | Service level of advertising break $[b_i^r, f_i^r]$, $i \in A$ and $r \in F$ |
| t_{ij} | The earliest available time on TV channel i in which a commercial j can be processed, $i \in A$ and $j \in B$ |
| The objective value | |
| L_{max} | The maximum lateness over all commercials |

2.1. Assumptions

- A system has m TV channels, which do not operate continuously; each channel is ready for processing only in certain advertising breaks.

- The start and finish times of each advertising break and the number of all advertising breaks are known in advance.
- At any time, a commercial can be assigned to a maximum of one TV channel, and a TV channel can execute a maximum of one commercial.
- Advertising breaks can be classified into different service levels, and each advertising break is characterized by only one type of service level in the system.
- No commercials are allowed to be preempted.
- Each commercial can be processed only on a specified set of TV channels.
- Each commercial has a specific service-level requirement to be broadcasted. For example, a commercial with service level k implies that the commercial must be broadcast in an advertising break with service level k or below.

2.2. Notations

We are given m TV channels and n commercials with the following notations (Table 1) throughout this paper.

3. A branch and bound algorithm

In this section, we propose a branch and bound algorithm to find an optimal solution for the studied problem. We introduce the proposed algorithm in the following three stages. First, we introduce the bounding scheme in Section 3.1. We extend the modified *LFJ* (least flexible job first)-based algorithm proposed by Liao [20] to construct an *LFJ/EDD* (earliest due date first) algorithm to find a feasible solution and use the result as the upper bound. Then, we apply a two-phase binary search algorithm, proposed by Sheen and Liao [21], to find an optimal solution for scheduling machine-dependent jobs on machines with availability constraints and use this result as the lower bound of the studied problem. Following this, we discuss the branching scheme in Section 3.2. Finally, we propose the complete branch and bound algorithm in Section 3.3.

3.1. Bounding scheme

We introduce the procedures for obtaining a lower bound \underline{LB} and an upper bound \overline{UB} of the maximum lateness.

3.1.1. The procedure of the upper bounding

Our proposed algorithm to find a feasible schedule extends the modified *LFJ* algorithm provided by Liao [20]. Liao's algorithm solved the problem with identical machines in the presence of machine availability for minimizing makespan in polynomial time, where makespan is defined as the total length of the schedule. However, this approach did not account for due date constraints and only incorporated unit processing time. Hence, we extend the modified *LFJ* algorithm to integrate the *LFJ* and *EDD* rules to find a feasible schedule that allocates commercials in advertising breaks that satisfy the service-level requirements. The algorithm derives an upper bound of the maximum lateness for the following problem formulation:

The LFJ/EDD rule algorithm

Step 0: (Initialize)

0.1) Initialize sets SL_j for all commercials to obtain the advertising breaks that are capable of broadcasting commercial j with service level k in the TV channels.

For $i \in A$ and $r \in F$,

If $r_j < f_i^r$ and $(f_i^r - b_i^r) \geq p_j$ are satisfied,

$SL_j = SL_j \cup [b_i^r, f_i^r]$, where $[b_i^r, f_i^r] \in \cup_{l=1}^{rs_j} IS_l$ and $i \in M_j$.

0.2) Let $J^c = \{1, 2, \dots, n\}$; $L_{max} = -M$.

Step 1: (Choose a least flexible commercial j^*)

Find a commercial j^* in J^c that satisfies the condition $|SL_{j^*}| = \min_{l \in J^c} \{|SL_l|\}$.

If two or more commercials have the same value of $\min_{l \in J^c} \{|SL_l|\}$, then select a commercial with

the earliest due date as commercial j^* . Break ties by choosing commercials arbitrarily.

Step 2: (Select the earliest starting time of advertising break $[b_i^r, f_i^r]$ in SL_{j^*})

If $SL_{j^*} \neq \emptyset$, set $k = rs_{j^*}$.

2.1) Select the advertising break $[b_i^r, f_i^r]$ with service level k that satisfies the condition: $\min\{b_i^r\}$ in SL_{j^*} . In the event of ties, choose the advertising break arbitrarily.

2.2) If no advertising break has been selected, then set $k = k - 1$ and repeat Step 2.1.

Otherwise, no feasible solution exists and stop the algorithm.

Step 3: (Schedule commercial j^* in the selected advertising break $[b_i^r, f_i^r]$)

3.1) $t_{j^*} = \max\{r_{j^*}, b_i^r\}$ and $C_{j^*} = \max\{r_{j^*}, b_i^r\} + p_{j^*}$.

3.2) $J^c = J^c \setminus \{j^*\}$.

3.3) If $C_{j^*} - d_{j^*} > L_{max}$, then set $L_{max} = C_{j^*} - d_{j^*}$.

Step 4: (Update the advertising break $[b_i^r, f_i^r]$ in SL_j , for $j \in J^c$)

4.1) For $j \in J^c$,

If $[b_i^r, f_i^r] \in SL_j$,

If $C_{j^*} = f_i^r$, set $f_i^r = t_{j^*}$;

Else If $b_i^r = t_{j^*}$, set $b_i^r = C_{j^*}$

Otherwise,

- Set $[b_i^{\alpha+1}, f_i^{\alpha+1}] = [b_i^\alpha, f_i^\alpha]$, for $\alpha = r + 1, \dots, N(i) + 1$.

- Set $b_i^{r+1} = C_{j^*}$; $f_i^{r+1} = f_i^r$; $f_i^r = t_{j^*}$.

- If $(f_i^{r+1} - b_i^{r+1}) < p_j$, set $SL_j = SL_j \setminus \{[b_i^{r+1}, f_i^{r+1}]\}$.

4.2) If $(f_i^r - b_i^r) < p_j$, set $SL_j = SL_j \setminus \{[b_i^r, f_i^r]\}$.

Step 5: If $J^c = \emptyset$, then output L_{max} ; otherwise, go to Step 1.

Assume that there are m TV channels and each channel is only available for processing within each of $N(i)$ specified advertising breaks $[b_i^r, f_i^r]$ with service level rs_i^r . There are n commercials to be scheduled. Each commercial j is associated with a release time r_j , a processing time p_j , a due date d_j , and a required service level rs_j . Commercial j can only be processed for a specific set SL_j . A commercial is selected to be scheduled when it belongs to the least flexible commercial with the earliest due date in the unscheduled commercial set J^c , namely, the commercial with the minimum value of $|SL_j|$. The selected job j will be allocated to $[b_i^r, f_i^r]$ as early as possible. The commercials continue being allocated to suitable advertising breaks until all commercials are fully scheduled or

there exists no feasible schedule. We derive an upper bound of the maximum lateness for the studied problem when the proposed algorithm obtains a feasible schedule.

Theorem 1. Assume n commercials and total $\sum_{i=1}^m N(i)$ advertising breaks in m TV channels, as well as h service levels in the TV advertising scheduling problem. The time complexity of the *LFJ/EDD* rule algorithm is $O(n^2 \sum_{i=1}^m N(i))$.

Proof. The worst-case complexity is analyzed as follows:

Step 0: Need $O(n \sum_{i=1}^m N(i))$ time for initialization.

0.1) $O(n \sum_{i=1}^m N(i))$ time for initializing sets SL_j .

0.2) $O(1)$ time.

Step 1: Need $O(n)$ time to choose least flexible commercial j^* .

Step 2: Need $O(\sum_{i=1}^m N(i))$ time at Step 2 since there are, at most, $\sum_{i=1}^m N(i)$ available advertising breaks in SL_j .

Step 3: $O(1)$ time.

3.1) $O(1)$ time.

3.2) $O(1)$ time.

3.3) $O(1)$ time.

Step 4: Need $O(n \sum_{i=1}^m N(i))$ time to update the $[b_i^r, f_i^r]$ in set SL_j , for $j \in J^c$.

Step 5: $O(1)$ time.

The algorithm needs to iterate Step 1 to Step 5 n times. As such, the worst-case complexity of the algorithm is dominated by Step 4 and is $O(n^2 \sum_{i=1}^m N(i))$. The proof is complete.

3.1.2. The procedure of the lower bounding

In this section, we derive the lower bound \underline{LB} of the maximum lateness of the studied problem by applying the two-phase binary search algorithm proposed by Sheen and Liao [21]. In Sheen and Liao's article [21], the machine scheduling problem had n pre-emptive jobs on m machines under machine availability constraints with the objective of minimizing maximum lateness. The polynomial time two-phase binary search algorithm solved the problem optimally by solving a series of maximum flow problems. The time complexity of this algorithm is $O((n + (2n + 2x)^3 \log(UB - LB)))$, where $x = \sum_{i=1}^m N(i)$. We omit the detail because the algorithm is obtained directly from the work of Sheen and Liao [21].

3.2. Branching scheme

Each node α in the search tree is defined by J_α , J_α^c , $L_{max}(\alpha)$, $\overline{UB}(\alpha)$, and $\underline{LB}(\alpha)$, which are defined as follows. J_α represents the scheduled commercials on the TV channel i for $i \in A$. J_α^c represents all commercials that are not yet scheduled. $L_{max}(\alpha)$ indicates the maximum lateness among the scheduled commercials in J_α . $\overline{UB}(\alpha)$ and $\underline{LB}(\alpha)$ denote the upper and lower bounds of node α , respectively.

There is only one root node at the top of the search tree. Let α be the root node. There are no commercials in the corresponding J_α , and J_α^c contains all commercials. In node α , the proposed *LFJ/EDD* rule algorithm is applied to calculate the value of $\overline{UB}(\alpha)$ based on the commercials in J_α^c . Then, a two-phase binary search algorithm is used to find the lower bound $\underline{LB}(\alpha)$. A newly created node, say β , branching out from node α is created by taking m commercials in J_α^c and arranging

these commercials into sequence J_β . This means that node β corresponds to a partial schedule with m commercials scheduled as compared to node α . In the following, we offer a proposition that aims to eliminate unnecessary node branching.

Proposition 1. Assume two commercials j and l . Job j cannot be processed before job l on machine i if the following condition is satisfied, where $j, l \in J^c, j \neq l$

$$t_{ij} > \min_{l \in J} (t_{il} + p_l) \quad (1)$$

Proof. This proposition is easy to prove. If job j satisfies this inequality, then selecting job l before job j on machine i does not increase the value C_j and L_{max} . Therefore, scheduling job j at the current position on machine i is not an optimal schedule. The proposition follows.

From Proposition 1, it is not necessary to consider every node generated from the branching process. If job j is scheduled before job l and satisfies Eq (1), the node should be eliminated rather than branched.

3.3. The proposed branching and bound algorithm

In this section, we develop a branch and bound algorithm, which follows the branching and bounding schemes stated above. In the branch and bound tree, the root node has no commercials scheduled. The upper and lower bounds of the root node are obtained by applying the procedures of the upper and lower bounding, respectively. During the branching process, each other node in the tree is generated when there is a commercial that is unscheduled. Each node has at most m commercials scheduled during the available advertising breaks on m TV channels. If Proposition 1 is satisfied, the node is eliminated. According to the branching process, the upper and lower bounds can also be used to eliminate nodes that are not an optimal solution. Each iteration of the algorithm can obtain a partial schedule with at most m commercials on m TV channels. The node is then checked and new nodes are generated by performing the branching scheme and the upper and lower bounding procedures. In the following, we propose a branch and bound algorithm for solving the problem optimally.

3.3.1. The proposed branch and bound algorithm

Step 0: Initialization

- 0.1) Set $q = 0$; $r(i)=1$ and $M(i)=\text{False}$, for $i \in A$.
- 0.2) Create a root node, $root$ of the search tree.
 - Set $J_{root} = \{ \}$ and $J_{root}^c = \{1, 2, \dots, n\}$.
 - Calculate the upper bound $\overline{UB}(root)$ and $L_{max}(root)$ for commercials in J_{root}^c by the LFJ/EDD rule algorithm.
- Set $UB = \overline{UB}(root) = L_{max}(root)$.
- Calculate the lower bound $\underline{LB}(root)$ by the two-phase binary search algorithm.
- If $\underline{LB}(root) = UB$, then stop the algorithm and output the optimal schedule.
- 0.3) Set node $\alpha = root$ node.

Step 1: If $J_\alpha^c \neq \emptyset$, then branch out child nodes of node α and set $q = q + 1$.

If $|J_\alpha^c| < m$, then take $(m - |J_\alpha^c|)$ artificial commercials into set J_α^c .

Create a child node β for each combination of m commercials in J_α^c and put these m commercials into sequence J_β .

1.1) For $i \in A$, let $[b_i^{r(i)}, f_i^{r(i)}]$ be an advertising break with service level k and commercial j in the i th position of sequence J_β .

- If $M(i) = \text{True}$, then remove commercial j in the i th position of J_β and replace it with artificial commercial X .
- If Proposition 1 is satisfied, then remove node β .

1.2) Set $J_\beta^c = J_\alpha^c \setminus \{J_\beta\}$; $L_{max}(\beta) = L_{max}(\alpha)$; $ACTIVE_\beta = \text{False}$.

1.3) For $i \in A$, let commercial j be scheduled in the i th position of sequence J_β .

- Set $t_j = \max\{r_j, b_i^{r(i)}\}$ and $C_j = \max\{r_j, b_i^{r(i)}\} + p_j$;

If $L_{max}(\beta) < C_j - d_j$, then set $L_{max}(\beta) = C_j - d_j$.

If $L_{max}(\beta) \geq UB$, then remove node β .

- Update the advertising break set of node β .

If $C_j = f_i^{r(i)}$, set $f_i^{r(i)} = t_j$ and $r(i) = r(i) + 1$.

Else If $b_i^{r(i)} = t_j$, set $b_i^{r(i)} = C_j$.

Otherwise,

- Set $[b_i^{\alpha+1}, f_i^{\alpha+1}] = [b_i^\alpha, f_i^\alpha]$ for $\alpha = r(i) + 1, \dots, N(i)$.

- $N(i) = N(i) + 1$.

- Set $b_i^{r(i)+1} = C_j$; $f_i^{r(i)+1} = f_i^{r(i)}$; $f_i^{r(i)} = t_j$.

- Set $r(i) = r(i) + 1$.

While $(f_i^{r(i)} - b_i^{r(i)}) < p_j$ and $r(i) \leq N(i)$, for all $j \in J_\beta^c \cap R_k$,

do $r(i) = r(i) + 1$;

if $(i) = N(i)$, then set $M(i) = \text{True}$.

1.4) Calculate the lower bound and the upper bound of node β .

- Calculate the upper bound $\overline{UB}(\beta)$ by the *LFJ/EDD* rule algorithm.

If $\overline{UB}(\beta) \leq UB$, then set $UB = \overline{UB}(\beta)$.

- Calculate the lower bound $\underline{LB}(\beta)$ by the two-phase binary search algorithm.

If $\underline{LB}(\beta) > UB$, then remove node β .

Step 2: If $q \neq 0$, then select the node with the lowest lower bound and its $ACTIVE_\alpha = \text{false}$ at level q .

2.1) Set node $\alpha =$ selected node; $ACTIVE_\alpha = \text{True}$, go to Step 1.

2.2) If no node with $Active = \text{False}$ exists at level q , then set $q = q - 1$ and repeat Step 2.

Otherwise, stop the algorithm and output the optimal schedule.

4. Computational analysis

Here, we discuss a computational analysis for the proposed branch and bound algorithm. To conduct the evaluation of the proposed algorithm, we code the branch and bound algorithm in Visual C++ and test it on a number of randomly generated problems. The computational experiments are run on a PC Intel Core(TM) i5-4570 3.2 GHz with 4 GB RAM. The test instances are generated according

to the following scheme. We select the values of parameters based on the information that we observe from TV stations in Taiwan. A TV station has three to eight different channels to broadcast different types of TV programs. In the given planning horizon, which is usually an hour, there are four to eight advertising breaks that can air commercials. Each advertising break has a length $[b_i^r, f_i^r]$ between 50 to 70 s and is categorized according to the higher, average, and lower TV rating points. Therefore, each advertising break is classified into the three types of service levels. For each commercial, the processing time p_j is set to 5, 10, 15, 20, 25 or 30 s, and a specific service-level requirement to be broadcasted is applied.

Table 2. Computational results of the proposed branch and bound algorithm.

| m | n | Number of nodes created | | | Computational time (sec.) | | | Avg. child nodes eliminated (%) |
|-----|-----|-------------------------|---------|------|---------------------------|---------|--------|---------------------------------|
| | | Max. | Average | Min. | Max. | Average | Min. | By LB, UB and Proposition 1 |
| 3 | 5 | 3 | 1.25 | 1 | 0.28 | 0.11 | 0.06 | 51.27 |
| | 10 | 7 | 2.35 | 1 | 2.23 | 0.47 | 0.14 | 50.41 |
| | 20 | 14 | 3.05 | 1 | 115.32 | 24.65 | 3.85 | 48.15 |
| | 30 | 28 | 5.60 | 1 | 740.57 | 203.91 | 24.70 | 45.89 |
| | 40 | 52 | 12.05 | 1 | 1008.48 | 488.59 | 91.20 | 44.75 |
| 5 | 5 | 5 | 1.60 | 1 | 0.67 | 0.23 | 0.14 | 45.18 |
| | 10 | 13 | 1.60 | 1 | 2.75 | 1.14 | 0.42 | 50.01 |
| | 20 | 13 | 2.15 | 1 | 118.90 | 27.80 | 4.52 | 50.00 |
| | 30 | 44 | 6.05 | 1 | 913.44 | 321.80 | 24.70 | 52.13 |
| | 40 | 98 | 13.50 | 1 | 1747.81 | 549.53 | 109.04 | 55.00 |
| 8 | 5 | 4 | 1.40 | 1 | 1.17 | 0.50 | 0.20 | 40.47 |
| | 10 | 21 | 8.05 | 1 | 8.70 | 3.31 | 1.07 | 45.00 |
| | 20 | 31 | 12.75 | 1 | 306.99 | 97.69 | 16.30 | 55.38 |
| | 30 | 367 | 47.10 | 1 | 10257.90 | 653.14 | 309.22 | 62.63 |
| | 40 | 645 | 70.70 | 1 | 17523.25 | 917.66 | 317.22 | 67.35 |

We set $n \in \{5, 10, 20, 30, 40\}$, $m \in \{3, 5, 8\}$, $r_{max} = 3000$, $p_{max} = 6$, $rs_{max} = 3$, $l_{max} = 600$, and $l_{min} = 50$. For each job j , the release time r_j , processing time p_j , due date d_j , and required service level rs_j are randomly drawn from uniform distributions on the intervals $[0, r_{max}]$, $[1, p_{max}] \times 5$, $[0, r_{max} + 600] + r_j + p_j$, and $[1, rs_{max}]$, respectively. Each TV station is capable of processing job j with a probability equal to 0.5. For each TV channel i , the number of advertising breaks $N(i)$ is randomly drawn from the discrete uniform distribution $[4, 8]$. For each advertising break r on TV channel i , the start time b_i^r and end time f_i^r , as well as the service level rs_i^r , are generated as follows.

- b_i^1 is randomly drawn from the uniform distribution $[0, l_{max}]$;
- b_i^r is randomly drawn from the uniform distribution $[0, l_{max}] + f_i^{r-1}$, $r \in F$;
- f_i^r is randomly drawn from the uniform distribution $[l_{min}, 70] + b_i^r$, $r \in F$.
- rs_i^r is randomly drawn from the uniform distribution $[1, rs_{max}]$, $i \in A$ and $r \in F$.

In total, there are 20 test instances for each combination of n and m tested. The computational results are presented in Table 2. The columns labelled Max. (Min.) indicate the maximum (minimum) number of nodes created and the maximum (minimum) computational time among the test instances, respectively. The columns labelled “Average” give the average number of nodes created and the average computational time, respectively. From Table 2, we can obtain optimal solutions using the branch and bound algorithm for the problem size of 40 commercials and eight TV channels within a moderate CPU time. The larger the problem size, the more nodes created, and the more computational time will be required.

5. Conclusions

In this paper, we presented an exact branch and bound algorithm to find an optimal solution for scheduling TV commercials to minimize the maximum lateness. The branch and bound algorithm incorporates the *LFJ* and *EDD* rules to find feasible schedules as upper bounds and lower bounds based on an algorithm that combines a binary search and a network flow technique for pruning nodes. In the studied problem, commercials had service-level constraints and processing times and could not be split at any point in time. This problem is more general than the scheduling problem discussed in [20] and [21]. A large number of nodes can be eliminated by upper and lower bounds, and only a very small percentage of potential nodes are generated. The branch and bound algorithm can be used to solve the problem optimally for up to 40 commercials and eight TV channels.

Acknowledgments

The author is grateful for the financial support from the Ministry of Science and Technology in Taiwan under grant 108-2622-E-025-001-CC3 and from National Taichung University of Science and Technology under grant NTCUST110-19.

Conflict of interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

1. O. Sinha, P. Dharmarha, P. Kumar, Application of goal programming to optimize the schedule and frequency of product advertisement in Telemedia, in *Recent Advances in Mechanical Engineering*, Springer, Singapore, (2021), 875–884. https://doi.org/10.1007/978-981-15-9678-0_73

2. M. Singh, M. Pant, A. Kaul, P. C. Jha, Advertisement scheduling models in television media: a review, *Soft Comput. Theor. Appl.*, **584** (2018), 505–514. https://doi.org/10.1007/978-981-10-5699-4_47
3. *Statista*, Broadcast television advertising revenue of NBCUniversal from 2012 to 2020, 2021. Available from: <https://www.statista.com/statistics/811648/nbcuniversal-broadcast-tv-ad-revenue>.
4. *Statista*, TV advertising revenue in the United States from 2019 to 2025, 2021. Available from: <https://www.statista.com/statistics/259974/tv-advertising-revenue-in-the-us/>.
5. F. G. Tari, R. Alaei, Scheduling TV commercials using genetic algorithms, *Int. J. Prod. Res.*, **51** (2013), 4921–4929. <https://doi.org/10.1080/00207543.2013.778431>
6. K. Czerniachowska, Scheduling TV advertisements via genetic algorithm, *Eur. J. Ind. Eng.*, **13** (2019), 81–116. <https://doi.org/10.1504/EJIE.2019.097926>
7. A. García-Villoria, S. Salhi, Scheduling commercial advertisements for television, *Int. J. Prod. Res.*, **53** (2015), 1198–1215. <https://doi.org/10.1080/00207543.2014.951095>
8. D. B. Fontes, P. A. Pereira, A. C. C. Fontes, A decision support system for TV self-promotion scheduling, *Int. J. Adv. Trends Comput. Sci. Eng.*, **8** (2019), 134–140. <https://doi.org/10.30534/ijatcse/2019/06822019>
9. F. Díaz-Núñez, N. Halman, Ó. C. Vásquez, The TV advertisements scheduling problem, *Optim. Lett.*, **13** (2019), 81–94. <https://doi.org/10.1007/s11590-018-1251-0>
10. A. R. Brown, Selling television time: An optimisation problem, *Comput. J.*, **12** (1969), 201–207. <https://doi.org/10.1093/comjnl/12.3.201>
11. K. Hägele, C. Ó Dúnlaing, S. Riis, The complexity of scheduling TV commercials, *Electron. Notes Theor. Comput. Sci.*, **40** (2001), 162–185. [https://doi.org/10.1016/S1571-0661\(05\)80043-X](https://doi.org/10.1016/S1571-0661(05)80043-X)
12. S. Bollapragada, H. Cheng, M. Phillips, M. Garbiras, M. Scholes, T. Gibbs, et al., NBC's optimization systems increase revenues and productivity, *Interfaces*, **32** (2002), 47–60. <https://doi.org/10.1287/inte.32.1.47.19>
13. S. Bollapragada, M. R. Bussieck, S. Mallik, Scheduling commercial videotapes in broadcast television, *Oper. Res.*, **52** (2004), 679–689. <https://doi.org/10.1287/opre.1040.0119>
14. M. J. Brusco, Scheduling advertising slots for television, *J. Oper. Soc.*, **59** (2008), 1363–1372. <https://doi.org/10.1057/palgrave.jors.2602481>
15. S. Bollapragada, M. Garbiras, Scheduling commercials on broadcast television, *Oper. Res.*, **52** (2004), 337–345. <https://doi.org/10.1287/opre.1030.0083>
16. D. R. Gaur, R. Krishnamurti, R. Kohli, Conflict resolution in the scheduling of television commercials, *Oper. Res.*, **57** (2009), 1098–1105. <https://doi.org/10.1287/opre.1080.0635>
17. F. Guerriero, G. Miglionico, F. Olivito, Managing TV commercials inventory in the Italian advertising market, *Int. J. Prod. Res.*, **54** (2016), 5499–5521. <https://doi.org/10.1080/00207543.2016.1162919>
18. Y. Ma, C. Chu, C. Zuo, A survey of scheduling with deterministic machine availability constraints, *Comput. Ind. Eng.*, **58** (2009), 199–211. <https://doi.org/10.1016/j.cie.2009.04.014>
19. G. Schmidt, Scheduling with limited machine availability, *Eur. J. Oper. Res.*, **121** (2000), 1–15. [https://doi.org/10.1016/S0377-2217\(98\)00367-1](https://doi.org/10.1016/S0377-2217(98)00367-1)
20. L. W. Liao, Scheduling jobs with service level requirements on parallel machines under availability and eligibility constraints, *JIOS*, **30** (2009), 1157–1176. <https://doi.org/10.1080/02522667.2009.10699933>

-
21. G. J. Sheen, L. W. Liao, Scheduling machine-dependent jobs to minimize lateness on machines with identical speed under availability constraints, *Comput. Oper. Res.*, **34** (2007), 2266–2278. <https://doi.org/10.1016/j.cor.2005.09.002>



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)