A Decision Support System for TV self-promotion Scheduling

Dalila B.M.M. Fontes[1], Paulo A. Pereira [2], Fernando A.C.C. Fontes [3]
[1] LIAAD-INESC L.A., Faculdade de Economia, Universidade do Porto,
4200-464 Porto, Portugal, fontes@fep.up.pt
[2] Departamento de Matemática e Aplicações, Universidade do Minho
4800-058 Guimarães, Portugal, papereira@math.uminho.pt
[3] Systec-ISR-Porto, Faculdade de Engenharia, Universidade do Porto,
4200-465 Porto, Portugal, faf@fe.up.pt

ABSTRACT

This paper describes a Decision Support System (DSS) that aims to plan and maintain the weekly self-promotion space for an over the air TV station. The self-promotion plan requires the assignment of several self-promotion advertisements to a given set of available time slots over a pre-specified planning period. The DSS consists of a data base, a statistic module, an optimization module, and a user interface. The input data is provided by the TV station and by an external audiometry company, which collects daily audience information. The statistical module provides estimates based on the data received from the audiometry company. The optimization module uses a genetic algorithm that can find good solutions quickly. The interface reports the solution and corresponding metrics and can also be used by the decision makers to manually change solutions and input data. Here, we report mainly on the optimization module, which uses a genetic algorithm (GA) to obtain solutions of good quality for realistic sized problem instances in a reasonable amount of time. The GA solution quality is assessed using the optimal solutions obtained by using a branch-and-bound based algorithm to solve instances of small size, for which optimality gaps below 1% are obtained.

Key words : Decision support system, genetic algorithms, scheduling, TV self-promotion.

1. INTRODUCTION

Over-the-air television stations commonly broadcast their shows for free to anyone who can receive the signal. Nevertheless, the television networks can generate revenues from advertisers who are willing to pay for commercial time slots inserted in between the shows [1]. Most of the income of broadcasting TV stations comes from selling commercial time slots, also referred to as breaks. Certain breaks, time of day and day of the week combinations, are more desirable to customers than others; in particular, customers are interested in advertising in breaks close to shows that have a higher audience in their target group. Self-promotion is very important for TV stations; for instance, in 2016, the four

Portuguese open TV channels broadcasted 123 298 self-promotion spots, which took up 997 hours. This means, that on average about 41 minutes a day were used by each channel to promote their shows. Furthermore, in that same year self-promotion accounted for about 15% of the total advertisement time (commercial and non-commercial).

TV stations make use of self-promotion time slots to advertise the shows to be broadcasted in order to increase the shows' audiences and thus, the value of the commercial breaks; which in turn leads to higher revenues from selling the latter breaks. The time made available for self-promotion is limited and thus, it needs to be effectively used. The effective use of time slots implies its effective scheduling, that is, the time sequence of shows to be advertised.

Optimization techniques have been extensively used in many different application areas; nevertheless, there are only a few studies that report optimization problems in the television and media industry. The majority of such reports focus on scheduling television shows and commercial advertisements and on modelling audience, as well as audience behavior.

Over the years many researchers addressed problems in scheduling TV commercials, starting with Brown 1969 [2]. Although these problems are closely related to the one being addressed here, they mainly different in their purpose. Scheduling TV commercials aims at allocating a group of advertisement time slots to advertisers such that revenue/total sales is maximized, while satisfying advertisers requirements; or choosing the advertisers to which the commercial spots are sold, from a large pool of bidding advertisers [3]-[10], not maximizing the overall contacts. Some of these problems have been proven NP-hard [11], [12]. The problem being addressed here is concerned with increasing the audience of the show to be broadcasted, that is the total number of contacts on the target audience, which is designated as gross rating points (GRPs).

The remainder of this paper is organized as follows. In Section 2, we describe the problem and provides its mathematical programming model. In Section 3, we propose the solution approach. Section 4 discusses the genetic algorithm. Section 5 reports computational results. In Section 6, some conclusions are drawn.

2. PROBLEM DESCRIPTION AND FORMULATION

TV channels divide on-air time between shows, commercial breaks, and self-promotion breaks. Here, we study the use of self-promotion breaks in order to maximize shows audience and therefore, the value of commercial breaks.

Future shows are promoted by broadcasting small advertisement campaigns (designated as spots) in one or more self-promotion breaks (here and hereafter referred to as breaks). Spots are characterized by a time duration; the show they advertise; the show nature, which imposes a time of the day in which it can be broadcasted; a target audience; and a minimum number of contacts with viewers of the aforementioned target. Breaks are characterized by the broadcasting time; the forecasted audience in each target; and a time duration. We wish to determine which spots should be broadcasted in each break in order to maximize the GRPs obtained with such advertisement; this way increasing show audience within the desired targets. (GRP is of the utmost importance since it represents the number of contacts between a spot and its targeted audience. Therefore, since this is the measure used to price commercial breaks, GRPs are maximized in our problem.)

The mathematical programming model (1) to (7) uses the notation given in Table 1.

TABLE 1: Mathematical Notation.

Sets and indices:
B - Set of breaks, indexed by i;
S - Set of spots, indexed by j;
B_j - Set of breaks with feasible airing time for spot $j \in S$.

Parameters:
c_{ij} - GRPs obtained at break $i \in B$ by spot $j \in S$;
d_j - duration of spot $j \in S$;
BD_i - maximum used duration of break $i \in B$;
Bd_i - minimum used duration of break $i \in B$;
SN_j - maximum number of uses of spot $j \in S$;
Sn_j - minimum number of uses of spot $j \in S$;
C_j - minimum number of GRPs required for spot $j \in S$.

Decision variables:
x_{ij} - binary variables set to 1 if spot $j \in S$ is advertised in break $i \in B$ and zero otherwise.

$$\text{Maximize} \quad \sum_{i \in B} \sum_{j \in S} c_{ij}\, x_{ij} \tag{1}$$

Subject to:

$$\sum_{j \in S} d_j\, x_{ij} \leq BD_i, \quad \text{for all } i \in B, \tag{2}$$

$$\sum_{j \in S} d_j\, x_{ij} \geq Bd_i, \quad \text{for all } i \in B, \tag{3}$$

$$\sum_{i \in B_j} x_{ij} \leq SN_j, \quad \text{for all } j \in S, \tag{4}$$

$$\sum_{i \in B_j} x_{ij} \geq Sn_j \quad \text{for all } j \in S, \tag{5}$$

$$\sum_{i \in B_j} c_{ij}\, x_{ij} \geq C_j, \quad \text{for all } j \in S, \tag{6}$$

$$x_{ij} \in \{0,1\}, \quad \text{for all } i \in B, j \in S. \tag{7}$$

Expression (1) defines the objective that is to maximize the total number of GRPs obtained by allocating existing spots to the available breaks. Constraints (2) and (3) impose, for each break, the total duration of advertisements aired to be within break defined limits; while constraints (4) and (5) enforce the number of times each spot is aired, in feasible breaks, to be within spot specified limits. Constraints (6) ensure a minimum GRPs per spot. Finally, constraints (8) impose the nature of the decision variables.

3. THE SOLUTION APPROACH: A DSS

Decision Support Systems (DSSs) have been utilizes to aid decision making in almost every areas of business. Here, we describe a DSS developed specifically for a Portuguese TV station. The developed DSS has four components: a data base, a statistic module, an optimization module, and an user interface.

The data base has information on: i) weekly available breaks; ii) weekly available spots; iii) weekly show requirements; and iv) daily rating and audience values for each target-break pair. The statistical module makes use of this information in order to estimate GRPs, needed by the optimization module to solve the problem described in Section 2.

The interface has several functionalities including inputting and updating data (Figure 1), outputting the solution found (figures 2 and 3), modifying the outputted solution (Figure 3), and analyzing/changing the performance metrics (Figure 4).
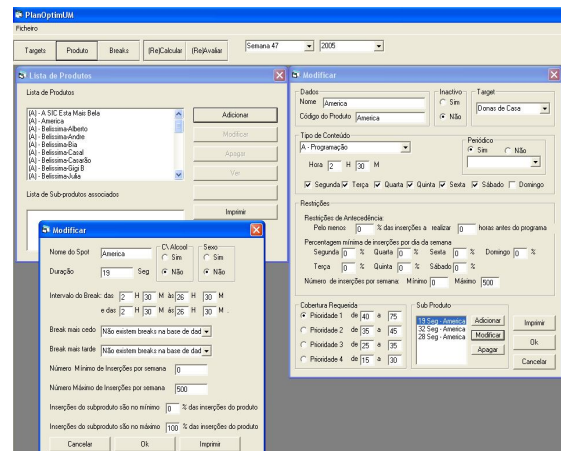

Figure 1: Interface to modify Advertisement characteristics.

Figure 2: A solution: Weekly schedule, list view

Figure 3: Monday schedule, grid view

Figure 4: Products and their indicators (GRP's and Coverages).

4. GENETIC ALGORTHM

The optimization module, mainly, consists of a genetic algorithm that is discussed next. GAs have been successfully employed in many application areas, see, e.g., [13], [14]. The optimization module also includes the solution of subproblems that allow for the transformation of the original

problem into the problem described in Section 2. For example, rating and audience values are given for target-break pairs and need to be converted into spot-break pairs. Similarly, requirements such GRPs, number of times advertised, and airing time are for shows not spots.

The DSS considers the problem with its full complexity, while the genetic algorithm is used with a simplified version of the problem. Therefore, as previously mentioned, some subproblems are solve and some approximations considered in order to obtain the model provided in Section II.

We propose a GA that ensures solution feasibility by employing a repair procedure whenever needed, in a similar fashion of that proposed in [15], [16].

4.1 Encoding and decoding

A chromosome consists of $N_B \times N_S$ genes, and is represented as a binary matrix, where N_B and N_s are the number of breaks and the number of spots, respectively. The matrix element (i, j) is set to 1, representing the airing of spot j int break i, or set to 0 otherwise.

If the solution obtained is not feasible then a repair procedure attempts to make it feasible by applying the three following phases in turn.

Phase I – DROP.
1. Find breaks whose duration is exceeded.
2. Consider them, one at the time, in descending order of excessive usage.
   While needed remove spots being aired by increasing values of benefit per unit of cost
   $$u_{ij} = \frac{c_{ij}}{d_j \times w_i},$$

where the benefit c_{ij} is measured as GRPs contribution of spot j when aired in break i and the cost is obtained by multiplying the duration d_j of spot j by the marginal value of time in break i, given by the dual variables w_i associated with constraints (2).

Phase II – ADD
1. Find spots for which the required GRPs are not met.
2. For each such spot, find breaks not airing it with enough unused time, at least d_j.
   While needed insert spots in decreasing order of c_{ij}.

Phase III – SWAP
1. Find spots for which the required GRPs are not met (say j).
2. For each such spot, find spots (say k) for which the required GRPs are over satisfied such that:
   2.1 The unused time of the break where spot k is aired plus the duration of spot k is at least the duration of spot j.
   2.2 Spot k still have enough GRPs if removed.
3. Remove spots k in increasing order of $\{c_{ik} - c_{ij}\}$.

4.2 Evolutionary strategy

We consider the population of the current iteration and the fitness value for each chromosome, i.e., total GRPs obtained,

and divide the population into two sets: the elite set, which consists of the best 10%, and the non-elite set, which has the remaining solutions. Following on the work in [17], the population for the next generation is obtained by i) copying the elite solutions of the current population, ii) randomly generating 5% of the population, and iii) obtaining the remaining solutions by crossover.

Parents are selected by Stochastic Universal Sampling (SUS) [18], one from the elite set, the other from the non-elite set. Crossover is accomplished via single point crossover. Starting by selecting one crossover point, the two offsprings are obtained by concatenating the binary string from the beginning of the chromosome of the first parent until the crossover point, and the remaining part is copied from the second parent.

5. NUMERICAL RESULTS

In order to test the proposed GA and infer on the quality of the solutions it finds, we have randomly generated a set of 60 problem instances, consisting of 5 instances of 12 different sizes. Problem size is represented as (a, b), where a is the number of breaks and b the number of spots. All parameters are uniformly distributes as follows: c_{ij} in [5,100], d_j in [6,60], C_j in [10,50], Sn_j in [0,1], SN_j in [1,3], $BD_i = D_i$ in [60,150], and Bd_i in [0, $D_i/2$].

Due to the stochastic nature of GAs, we run the proposed GA 10 times for each problem instance. The solutions obtained by the GA were compared to optimal ones obtained by a branch-and-bound. Figure 5 reports on the GA average optimality gap calculate as in (8) and Figure 6 on the GA average computational time required to find such solutions. The averages reported refer to the 10 runs of each of the five problem instances solved for each of the 12 instance sizes.

$$Gap = \frac{GA - Opt}{Opt} \times 100,\qquad(8)$$

where, Opt and GA represent, respectively, the total GRPs obtained by an optimal solution and by a GA solution.
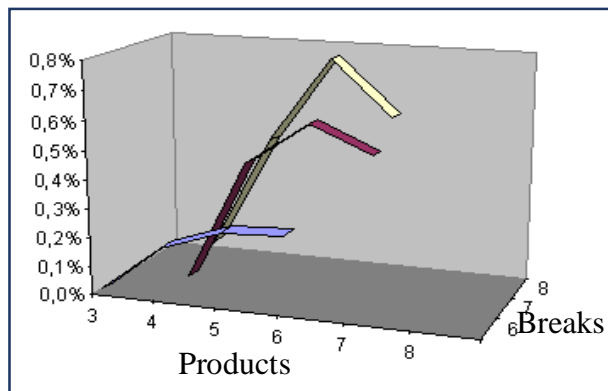

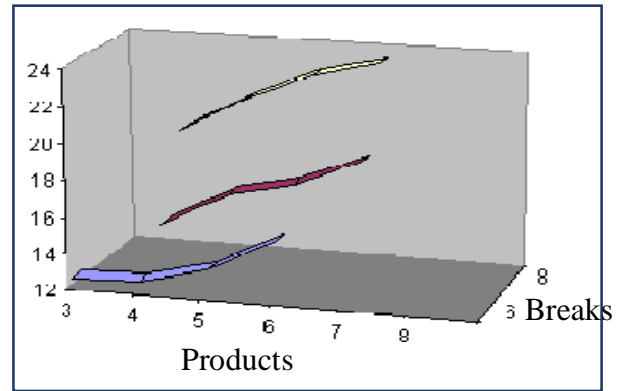**Figure 5:** Optimality gap for randomly generated problem instances.

Figure 6: Computational time for randomly generated problem instances.

As it can be seen, the GA finds very good quality solutions quickly. Indeed, the optimality gap is always below 1% with computational time requirements never exceeding 24 seconds. However, the test instances used are much smaller than the ones the company needs to solve. Therefore, we have also used the GA on real instances and compared the solutions obtained with those of the company's current practice.

The company use to do the planning manually and one dedicate operator would spend one full week preparing the following week schedule. Given the time required no new information, for example cancelled shows, could be account for.

The weekly problem encountered by the TV station, requires the broadcast of about 1350 spots (of 50 different shows, approximately), chosen from around 90 different spot instances, in arout 230 breaks. We found solutions for 10 instances of this size by running the GA until stagnation, which occurred on average after about 5 hours. Although 5 hours seems to be a large value, it represents a huge improvement when compared with company's practice. In addition, the GA can be run without supervision, for example overnight, allowing the operator to concentrate solely on changing the schedule obtained either due to new information or due to operator's informal knowledge, which is not accounted for.

Regarding solution quality, although the total number of spots aired was about the same, the GA can find solutions with an increase total number of GRPs, about 37% higher. We also tested the GA for short runs and were able to find out that it can find solutions with a number of GRPs similar to that of company's current practice in about 20 minutes.

6. CONCLUSION

We consider a new problem faced by TV stations, the TV Self-promotion Scheduling Problem. Since the original problem is too complex a simplified version has been considered for developing a genetic algorithm. The simplified version, however, retains the important features of the original problem and is itself a new combinatorial optimization problem. It can been seen that the problem considered has similarities with other well-known combinatorial optimization problems, namely the multiple knapsack

problem, while having some distinctive features that characterize it as a distinct new problem.

While our main interest was to solve problem instances of large-size (the ones faced by the TV station), we generated small-sized instances in order to more easily test the quality of the solutions obtained. For these smaller-sized instances, the generated solutions whose value is within 1% of the optimal solution value, on average.

A decision support system (DSS) has been developed to address this problem. The developed DSS was tailored for an over-the-air commercial television that frequently leads audience shares in Portugal.

On average there are about 50 shows to be advertised each week and the number of different spots to advertise each one of them varies between 1 to 5. On one single week there are about 230 self-promotion breaks in which spots are to be scheduled.

Before the proposed solution was implemented, about a week to 10 days before a show would air, scheduling personnel used to allocate the self-promotion slots to the available breaks. The process of allocating slots to breaks was traditionally done manually and it would take about one full week to a dedicated operator to find a good one.

When comparing three months' observations on manually generated schedules and also on the schedules generated by the proposed DSS, we observe a significant qualitative improvement in the total number of GRPs obtained (about 37%). Additionally, similar quality schedules could be obtained in about 20 minutes.

ACKNOWLEDGMENT

REFERENCES

1. K. E. Kendall, J. E. Kendall, and E. J. Kendall. **Systems Analysis and Design**, Englewood Cliffs, NJ: Prentice Hall, 1992.
2. R. Brown. Selling Television Time: **An Optimisation Problem**, *The Computer Journal*, vol. 12, pp. 201-207, 1969.
   https://doi.org/10.1093/comjnl/12.3.201
3. Kimms and M. Muller-Bungart. **Revenue Management for Broadcasting Commercials: The Channel's Problem of Selecting and Scheduling the Advertisements to Be Aired**, *International Journal of Revenue Management*, vol. 1, pp. 28-44, 2007.
   https://doi.org/10.1504/IJRM.2007.011192
4. D. Gaur, R. Krishnamurti, and R. Kohli. **Conflict Resolution in the Scheduling of Television Commercials**, *Operations Research*, vol. 57, pp. 1098-1105, 2009.
   https://doi.org/10.1287/opre.1080.0635
5. M. S. Wuang, C. L. Yang, R. H. Huang, and S. P. Chuang. **Scheduling of Television Commercials**, in *Proc. IEEE Conf. on Industrial Engineering and Engineering Management*, 2010, pp. 803-807.
   https://doi.org/10.1109/IEEM.2010.5674192
6. X. Zhang. **Mathematical Models for the Television Advertising Allocation Problem**, *International Journal of Operational Research*, vol. 1, pp. 302-322, 2010.
   https://doi.org/10.1504/IJOR.2006.009303
7. J. Mao, J. Shi, J. Wanitwattanakosol, and S. H. Watanabe. **An ACO-Based Algorithm for Optimizing the Revenue of TV Advertisement Using Credit Information**, *International Journal of Revenue Management*, vol. 5, pp. 109-120, 2011.
   https://doi.org/10.1504/IJRM.2011.040304
8. R. Alaei and F. Ghassemi-Tari. **Development of a Genetic Algorithm for Advertising Time Allocation Problems**, *Journal of Industrial and Systems Engineering*, vol. 4, pp. 245-255, 2011.
9. F. Ghassemi-Tari and R. Alaei. **Scheduling TV commercials using genetic algorithms**, *International Journal of Production Research*, vol. 51, pp. 4921-4929, 2013.
   https://doi.org/10.1080/00207543.2013.778431
10. García-Villoria and S. Salhi. **Scheduling commercial advertisements for television**, *International journal of production research*, vol. 53, pp. 1198-1215, 2015.
    https://doi.org/10.1080/00207543.2014.951095
11. K. Hägele, C. O. Dunlaing, and S. Riis. **The Complexity of Scheduling TV Commercials**, *Electronic Notes in Theoretical Computer Science*, vol. 40, pp. 162-185, 2001.
    https://doi.org/10.1016/S1571-0661(05)80043-X
12. F. Díaz-Núñez, N. Halman, and O. C. Vásquez. **The TV advertisements scheduling problem**, *Optimization Letters*, vol. 13, pp. 81-94, 2019.
    https://doi.org/10.1007/s11590-018-1251-0
13. D. B. M. M. Fontes, J. F. Gonçalves, and F. A. C. C. Fontes. **An Evolutionary Approach to the Maximum Edge Weight Clique Problem**. *Recent Advances in Electrical & Electronic Engineering*, vol. 11, pp. 206-266, 2018.
    https://doi.org/10.2174/2352096511666180214105643
14. P. Srikanth, M. Srija, S. Chakravarthy, G. V. Rao, and G. A. Raju. **Compare various circuits area reduction using genetic algorithm and hybrid partitioning algorithm**. *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 7, pp. 111-114, 2018.
    https://doi.org/10.30534/ijatcse/2018/08762018
15. L. A. C. Roque, D. B. M. M. Fontes, and F. A. C. C. Fontes. **A biased random key genetic algorithm approach for unit commitment problem**, *Lecture Notes in Computer Science*, vol. 6630, pp. 327-339, 2011.
    https://doi.org/10.1007/978-3-642-20662-7_28
16. L. A. C. Roque, D. B. M. M. Fontes, and F. A. C. C. Fontes. **A hybrid biased random key genetic algorithm approach for the unit commitment problem**, *Journal of Combinatorial Optimization*, vol. 28, pp. 140-166, 2014.
    https://doi.org/10.1007/s10878-014-9710-8

17. J. F. Gonçalves and M. Resende. **Biased random-key genetic algorithms for combinatorial optimization**, *Journal of Heuristics*, vol. 17, pp. 487-525, 2010. https://doi.org/10.1007/s10732-010-9143-1

18. J. E. Baker. **Reducing Bias and Inefficiency in The Selection Algorithms**, in *Proc. 2nd International Conf. on Genetic Algorithms*, 1987, pp. 14-21.

AUTHOR PROFILE

Dalila B.M.M. Fontes is Assistant Professor, with Habilitation, at Faculty of Economics, University of Porto, Portugal and researcher in INESCTEC. She received the first degree in Electrical and Computer Engineering from the University of Porto, the M.Sc. in Operational Research from The London School of Economics and Political Sciences, the Ph.D. in Management Sciences from The Management School - Imperial College London, U.K. In 2011 she received the Habilitation degree (Agregação) in Management Sciences from the University of Porto.

Her research interests are in combinatorial optimization, having a specific interest in scheduling problems, network problems, genetic algorithms and other metaheuristics, linear programming and dynamic programming.

Paulo A. Pereira is Assistant Professor in the Department of Mathematics of University of Minho, Portugal. He received his PhD degree in Mathematics from University of Minho in 2009. His research interests are in the area of optimization, statistics and operational research.

Fernando A.C.C. Fontes is Associate Professor, with Habilitation, in the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Portugal and researcher in Systec - Institute of Systems and Robotics – Porto (ISR).

He received the first degree in Electrical and Computer Engineering from the University of Porto, the M.Sc. in Control Systems and the Ph.D. degrees from the Department of Electrical and Electronic Engineering, Imperial College London, U.K. In 2014 he received the Habilitation degree (Agregação) in Electrical and Computer Engineering from the University of Porto.

His research interests are in optimization and control theory, having a specific interest in nonlinear and constrained problems, optimal control, and model predictive control. His main scientific contributions are in model predictive control (stability and robustness conditions for nonlinear and sampled-data systems), in optimal control (stronger forms of the maximum principle and numerical methods) and in nonlinear optimization methods (dynamic programming based and other global optimization algorithms).

Homepage: http://www.fe.up.pt/~faf/