A Genetic Algorithm Approach for the TV Self-Promotion Assignment Problem

Paulo A. Pereira*, Fernando A. C. C. Fontes† and Dalila B. M. M. Fontes**

CMAT and Dept. of Mathematics for Science and Technology, University of Minho
4800-058 Guimarães, Portugal, `ppereira@mct.uminho.pt`

†*ISR and Faculdade de Engenharia, Universidade do Porto*
4200-465 Porto, Portugal, `faf@fe.up.pt`

**LIAAD-INESC Porto L.A. and Faculdade de Economia, Universidade do Porto*
4200-464 Porto, Portugal, `fontes@fep.up.pt`

Abstract. We report on the development of a Genetic Algorithm (GA), which has been integrated into a Decision Support System to plan the best assignment of the weekly self-promotion space for a TV station. The problem addressed consists on deciding which shows to advertise and when such that the number of viewers, of an intended group or target, is maximized. The GA proposed incorporates a greedy heuristic to find good initial solutions. These solutions, as well as the solutions later obtained through the use of the GA, go then through a repair procedure. This is used with two objectives, which are addressed in turn. Firstly, it checks the solution feasibility and if unfeasible it is fixed by removing some shows. Secondly, it tries to improve the solution by adding some extra shows. Since the problem faced by the commercial TV station is too big and has too many features it cannot be solved exactly. Therefore, in order to test the quality of the solutions provided by the proposed GA we have randomly generated some smaller problem instances. For these problems we have obtained solutions on average within 1% of the optimal solution value.

Keywords: Genetic Algorithms, Combinatorial Optimization, TV Self-Promotion Assignment Problem.
PACS: 02.10.Ox, 02.60.Pn

INTRODUCTION

We report on the development of an optimizer based on genetic algorithms which is integrated into a Decision Support System (DSS) to plan the best assignment of the weekly self-promotion space for a TV station. This project was developed for SIC, a commercial TV station that frequently leads audience shares in Portugal.

Advertising is the main source of income in commercial TV's and advertisement revenues grow with audiences. Therefore TV stations typically reserve a promotion space (broadcasting time) for self-promotion (future programs, etc.). However, the total broadcasting time dedicated to advertisement, both commercial and self-promotion, is regulated and limited. Hence, the self-promotion time should be kept small, thus optimizing its usage is of upmost importance. The problem consists on, given the weekly self-promotion space (a set of breaks with known duration) and a set of products to promote, assigning the products to the breaks in the "best" possible way. For each product we have a given target audience, and for each break we can estimate how many viewers of each target are watching. Moreover, each product has several assignment constraints to be satisfied. The objective is to maximize contacts in the target audience for each product, while satisfying all constraints.

The weekly problem faced by SIC, typically involves the broadcast of about 1350 spots, chosen from around 90 different ones, in roughly 200 time slots. The duration of each of these time slots, here and hereafter designated by breaks, ranges from 6 to 150 seconds. Given the problem complexity and dimensionality (see [1, 2] for a more detailed problem description), only heuristic approaches are expected to obtain solutions in a reasonable amount of time. In the next section, we describe a GA that also incorporates a greedy heuristic to initialize part of the population with "good solutions". While the DSS considers the original problem with all the complexity features, the GA works on a simplified version of the problem that, nevertheless, maintains the essential features of the problem. We have noticed that the simplified problem is a binary programming problem that has similarities with other known combinatorial optimization problems, such as the multiple knapsack problem (see e.g. [3, 4]).

The problem can be stated as follows. Let $i = 1, \ldots, N_B$ be the set of breaks, and $j = 1, \ldots, N_S$ the set of spots. Each break i and spot j are characterized by their durations B_i and d_j, respectively. Also, let c_{ij} denote the number of viewers that watch spot j when inserted in break i. We wish to find a spot-break assignment such that a minimum break usage b_i, for break i and a minimum number of viewers C_j, not necessarily different, for spot j are guaranteed. Amongst all possible assignments we are interested on the ones that maximize the number of viewers, from the intended target, watching the spots. This problem can be formulated as given in the following binary programming model.

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{i=1}^{N_B} \sum_{j=1}^{N_S} c_{ij} x_{ij} \\
\text{Subject to:} \quad & \sum_{j=1}^{N_S} d_j x_{ij} \leq B_i \quad \forall \ 1 \leq i \leq N_B, \\
& \sum_{j=1}^{N_S} d_j x_{ij} \geq b_i \quad \forall \ 1 \leq i \leq N_B, \\
& \sum_{i=1}^{N_B} c_{ij} x_{ij} \geq C_j \quad \forall \ 1 \leq j \leq N_S, \\
& x_{ij} \in \{0,1\} \quad \forall \ 1 \leq i \leq N_B \ \text{e} \ \forall \ 1 \leq j \leq N_S,
\end{aligned}
\tag{1}
$$

where the decision variables x_{ij} assume the value 1 if spot j is assigned to break i and 0 otherwise.

THE PROPOSED GENETIC ALGORITHM

Since Holland first proposed GAs in the early 1970s as computer programs that mimic the evolutionary processes in nature [5], the GAs have been demonstrating their power by successfully being applied to many practical optimization problems in the last decade.

We propose a GA that uses a greedy heuristic followed by a repair procedure to generate a population of feasible solutions. This population is then evolved through reproduction, crossover, and mutation. The reproduction and crossover operators determine which parents will have offspring, and how genetic material is exchanged between parents. The stochastic universal sampling (SUS) technique, introduced by [6], is used to select solutions for recombination. Such a selection is proportional to the fitness and exhibits no bias and minimal spread. SUS uses a single random value to sample all of the solutions by choosing them at evenly spaced intervals. Crossover happens between a solutions which is located in an even position with the solution located in the adjacent odd position. The single point crossover is used. After selecting one crossover point, two children are obtained by using the binary string from beginning of chromosome to the crossover point from one parent, the rest being copied from the other parent. An elitist strategy is used, since some of the best individuals are copied from one generation to the next. This is important since this way it is guaranteed that the best solution is monotonically improving from one generation to the next. However, such a strategy tends to increase the convergence rate. Therefore, and in order to avoid it mutation is also incorporated, through immigration, to allow for randomly introduction of new genetic material.

Greedy heuristic. In order to generate solutions we have implemented a constructive greedy heuristic procedure that outputs spot-break assignment binary matrices with good coverage.

Using the minimum number of viewers C_j required for each spot j, we start by estimating the maximum Mb_j and the minimum mb_j number of times that spot j may be used. Then spots are iteratively assigned to breaks, starting with the spot having the highest number of minimum times still to be used (initially mb_j). For such a spot we chose, the break with the larger audience rate for the targeted of the selected spot, that still has enough unused time. This is repeated until the coverage C_j has been provided for all spots, or the spot has already been used Mb_j times.

The repair routine. In addition and in order to guarantee the satisfaction of the constraints of problem, (mainly the first group of constraints that imposes not exceeding the breaks duration) we used a "repair" procedure, inspired in that of Chu and Beasley [7] for a multiple knapsack problem. This procedure has two phases: in the first one we guarantee that the break time is not exceeded by removing spots, whenever needed, while in the second phase we try to satisfy spot's number of viewers, or improve the solution, by adding spots to the breaks which have some unused time.

We start by selecting the most violated break, i.e. with the highest used time above its duration, and then iteratively remove from its spots in ascending order of u_{ij} where

$$u_{ij} = \frac{c_{ij}}{w_i d_j}$$

and w_i dual value of the break constraint (obtained by solving a linear programming relaxation of (1)). At the end of this phase we have a feasible solution. Then, spots with a number of viewers smaller than C_j and further inserted into breaks with unused time, in ascending order of time usage benefit, i.e. obtained as the product of the spot duration by the dual variable of the break constraint ($w_i d_j$). If this not possible swaps with over-satisfied spots are attempted.

COMPUTATIONAL RESULTS

In order to test the proposed GA we have used 60 randomly generated problems, 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. The GA was used 10 times for each of these problems, therefore in the following results each reported figure corresponds to the average value obtained from the 10 runs of the 5 problem instances. In Figure 1 we report on the solution quality and computational time requirements.
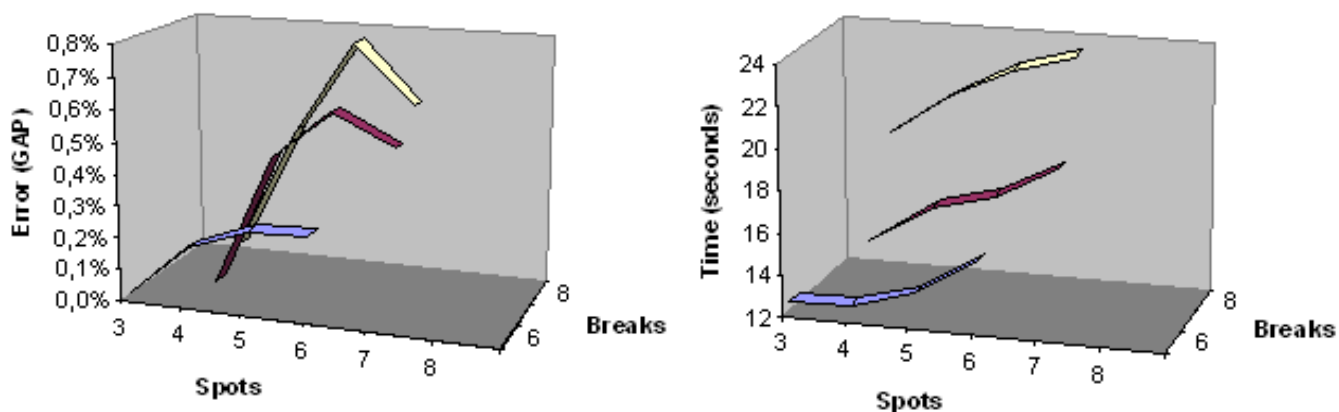

FIGURE 1. Computational performance for randomly generated problems.

The time requirements are reported in seconds and the solution quality is the optimality gap obtained as $\frac{GA_{sol} - OPT}{OPT}$, where OPT is the optimal solution value obtained by solving the problems using a branch-and-bound algorithm.

We have also solved 3 problem instances with 200 breaks and 50 spots, which is the size of the TV station problems. However, for such problems we can only report on computational time requirements, see Table 1, since the branch-and-bound algorithm only solves problems with up to 8 breaks. Again the results reported refer to the average of the 10 runs performed using the GA.

TABLE 1. Computational performance for real sized problems.

Size (200, 50)	Best Sol. Value	Best Sol. Generation	Best Sol. Time
Problem 1	107962.30	353.35	20692.30
Problem 2	112203.30	398.35	20791.30
Problem 3	107803.00	352.70	20782.80
Average		368.13	20755.50
Std. Dev.		21.40	44.80

CONCLUSIONS

We consider here a new type of combinatorial optimization problem, motivated by, and a simplification of, the TV Self-promotion Assignment Problem. This problem has similarities with other known combinatorial optimization problems, such as the assignment problem or the multiple knapsack problem, but has some distinctive features that caracterize it as a new problem. A genetic algorithm was developed to address this problem. The algorithm incorporates a greedy heuristic to initialize part of the population and uses a repair routine to guarantee feasibility of each member of the population. Although we were mainly interested in solving problems of high dimension (200 breaks by 50 spots), the quality of the solution was tested for smaller dimension problems for which we could find the exact global minimum using a branch-and-bound algorithm. For these smaller dimension problems we have obtained solutions on average within 1% of the optimal solution value.

REFERENCES

1. P. Pereira, F. Fontes, and D. Fontes, "A Decision Support System for Planning Promotion Time Slots," in *Operations Research Proceedings 2007: Selected Papers of the Annual International Conference of the German Operations Research Society (Gor)*, Springer, 2008, p. 147.
2. P. A. Pereira, *Um Sistema de Apoio à Decisão Baseado em Optimização para o Planeamento de Auto-promoção de uma Estação de Televisão*, Ph.D. thesis, Departamento de Matemática para a Ciência e Tecnologia, Universidade do Minho, Guimarães, Portugal (2009).
3. J. Tavares, F. Pereira, and E. Costa, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **38**, 604–616 (2008).
4. E. Lin, *INFOR* **36**, 274–317 (1998).
5. J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
6. J. Baker, "Reducing Bias and Inefficiency in The Selection Algorithms," in *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates Inc., USA, 1987, pp. 14–21.
7. J. Chu, P.C. e Beasley, *Journal of Heuristics* **4**, 63–86 (1998).