INTERFACES
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0092-2102 | EISSN 1526-551X | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Lessons Learned from 15 Years of Operations Research for French TV Channel TF1

Bouygues' corporate operations research team (the "Bouygues e-lab") has been working with this industrial group's various subsidiaries, including the leading French TV Channel TF1, for the past 15 years. This article presents the achievements obtained with this advertisement broadcaster and tries to identify the practical keys to success in this partnership. In particular, best practices for managing OR projects are pointed out and explained. The selected projects cover the internet and television businesses. They gave TF1 a competitive advantage by allowing them to provide quicker and better answers to advertisers' requests, in addition to a better usage of its limited and perishable airtime inventory. The induced increase of revenue is estimated by TF1 at €20 million per year.

Key words: media & advertisement; sales & marketing; revenue management; OR project management.
History:

1. Introduction

Bouygues' corporate operations research team (the "e-lab") has been working for 15 years with the various subsidiaries of this diversified industrial group. This small team (five OR engineers) reports directly to Bouygues' Chief Information Officer and acts as an analytics services company within the group. The group itself focuses on construction (building, civil works and electrical contracting, property development), telecoms, and media. In particular, a long-term partnership with TF1 led to a series of achievements in the advertisement business. The examples discussed in this paper are taken from work with TF1.

As a way of background, TF1 is a major media group in Europe and the leading television group in France, with almost 25 % audience share and 45 % of the TV advertising market. It operates 16 television channels, 128 radio stations, and 15 websites, generating a total of €2.4 billion in sales in 2009 (including €1.4 billion for the TF1 core channel). TF1's advertising arm sells time to advertisers. Television ads are sold respecting France's legal limit of 12 minutes per hour and 144 minutes a day. Ad slots are marketed according to customer segments targeted by advertisers (women under 50, men 15-49, etc.) and the estimated viewing audience. On TV, viewership is expressed in Gross Rating Points (GRP). GRP measure the audience reached by an advertisement campaign: 16 GRP stands for 16 % of the audience target. For instance, there are 14 million males aged 15 to 49 in France, hence 16 GRP for this marketing target amounts to 2.2 million viewers. TV advertisers buy a a certain number of seconds during a commercial break. The price per second of such a "slot" depends on the expected viewing audience, on the part of the day and on the negotiated discount rate of the advertiser. Such slots can be bought either one by one or as a package whose scheduling is computed by TF1 as a marketing planning service. The interested reader can find further details related to the TV advertising market in Gabszewicz et al. (1999) or in Goettler and Shachar (2001). Internet advertising is sold somewhat differently. The audience is often measured in terms of the number of visitors per day, and the ability for the visitor to click on advertisements on a webpage led to two different payment mechanisms: pay-per-view and pay-per-click. In the first model the advertising company pays a fixed fee each time its message is displayed to an internet visitor, whereas in the pay-per-click model, the payment is proportional to the number of times a visitor clicked on the advertisement.

In this article, we present three optimization problems encountered in the advertising business. First we illustrate our mathematical consulting activities with an analysis of internet click rates in a pay-per-click context, leading to a bandit arm algorithm. Then we describe a goal programming model which is used daily by the sales front office to build high-quality marketing plans for TV advertisers. Finally, the global optimization problem for the group's main channel is presented: an objective function that runs into the hundreds of millions of euros to be optimized every other month. Throughout the description of these projects we will attempt to identify the keys to success and eventually draw conclusions on the management of operations research projects and on the importance of Information Technologies (IT) issues in this context.

2. Mathematical Consulting: Optimizing Internet Click Rates

2.1. Business Context

Various mathematical questions arise in the advertisement business. For instance when advertising on internet sites, the probability for a user to click on a given advertisement is a crucial figure. On TF1 websites (17 million unique visitors per month), the so-called "shopping-boxes" follow this pay-per-click model. These shopping-boxes are advertisement messages presenting a product rather than a brand, with a direct link to a page where the internet user can buy it. Each advertising company provides a catalog of products and TF1 can freely select displayed products from this catalog. The total number of products is close to 20,000. Each time a page is displayed, the ad server should select the product with the highest click probability. Since only clicks are paid, this ad selection algorithm must be carefully designed.

2.2. Optimization and Difficulty

A statistical analysis of historical data allows key factors impacting the click rate of an advertisement for the product to be identified. These factors include the day of the week, (e.g. Friday), the time slot (e.g. from 5:00 PM to 6:59 PM), the product category (e.g. travel), the price level (e.g. from €25 to €50), and the internet site section (e.g. news, sports, cooking, etc.). A simple ad selection algorithm can be based on these facts. When selecting a product to display on a webpage at a certain time, the ad server can use such data to pick a product with the appropriate price and category. However, the "attractiveness" of the product itself appears to be the most important criterion, that is to say that similar products can have dramatically different click rates, possibly because the product is better or better advertised. Roughly speaking, we have to learn on the fly whether the blue swimsuit has a higher click rate than the red swimsuit, for example.

2.3. Solution Technique

This choice will be based on the observed clicking frequency of each product; however, the accuracy of these figures depends on the number of times a product was displayed. Here we are facing a classic exploitation vs. exploration dilemma: shall we display the product having the highest observed click rate or rather a product which was rarely presented in order to see if its attractiveness is even higher? In the literature, this problem is known as the multi-armed bandit problem. Given a number of bandit arms with unknown reward probabilities how do we maximize the cumulative reward obtained with a fixed number of coins? A powerful strategy for this kind of game consists in using the Upper Confidence Bound introduced by Auer et al. (2002), i.e. the upper bound of a confidence interval around the observed click frequency. Indeed products presenting a high upper confidence bound are either those showing a high click frequency over a large number of displays (narrow confidence interval around a high click frequency) or those rarely presented yet (wide confidence interval possibly around a poor click frequency). In other words this criterion judiciously combines the exploitation and exploration aspects. Note that Pandey and Olston (2007) reports significant gains by using this kind of strategy on Yahoo! Query logs. See also the work of Nakamura and Abe (2005) on the use of linear programming in this context (NEC Corporation).

3. Front-Office Software: Optimizing Advertisers' Plans

3.1. Business Context

In addition to its general interest channel, TF1 also operates theme channels that have become key components of the satellite, broadband, DTT, and cable networks in France. For these channels, advertisement slots are not sold one by one but as packages, also known as plans. For instance, a customer, that is to say an advertising company, may ask TF1 to build a "gold" plan for 30-second commercials on Eurosport, during the third week of November, with a budget of €20,000. The key aspect of this offer lies in the audience guarantee attached to these plans. Even if the advertiser does not select advertisement slots itself, it knows that the collective audience will reach a contractual minimum of say 16 GRP (recall that GRPs measure the number of viewers of an advertisement campaign). More precisely, this transcription of a budget into an audience depends on the selected "shape" of the package. For instance, a "gold" plan will have 60 % of GRP scheduled during primetime hours and a cost of €800 per GRP. For TF1, building such a plan amounts to solving a multi-dimensional knapsack problem where a certain number of slots have to be selected among available ones in such a way that the sum of prices is smaller than €20,000 and the sum of GRP is larger than 16, while 60 % of these GRP come from primetime slots. Note that Bollapragada et al. (2002) report a related optimization problem for the NBC channel.

3.2. Optimization and Difficulty

Solving this knapsack problem by hand is a complex and tedious task for media planners, who are responsible for building and selling advertisement plans to advertisers. Hence we developed a preliminary integer program to automatically compute such plans, based on these general conditions of sales. After a few days of work we were able to present our first plans using an Excel-based visualization. Prototyping is never a waste of time in OR. In the present case it revealed that the actual problem was far more complex. Most of our plans were rejected by experts because they presented patterns unacceptable to our customers: commercials in consecutive TV-breaks, an unbalanced dispatch of commercials among days of the week, repeated broadcasting hours on consecutive days (risk that the commercial will be seen by the same viewers every day), etc. In fact these side-constraints were so implicit in the practice of advertisement planning that they had been omitted in the specifications. Besides, we understood that almost all constraints are soft, meaning that a proposal must be tailor-made for our customer, possibly for a smaller budget or by relaxing some dispatching constraints. This would be the case if airing time inventory is saturated or customer requirements are conflicting. For instance, when launching a new product the advertiser might specify that it wants 60 % of slots on the first days of the plan, which contradicts balancing rules. Finally, optimizing revenue requires minimizing losses due to small residual durations in TV breaks: if we assign a 25-second commercial to a 30-second TV break, the residual 5 seconds will be difficult to sell. Using statistics on advertisement durations, this cost can be estimated and taken into account.

3.3. Solution Technique

As a result, our resolution scheme is based on goal programming (GP) modeling of these constraints, with an intensive use of classical GP techniques like lexicographic orders, weighting, satisfaction intervals, min-max, min-sum (Kornbluth 1973, Ignizio 1983). The majority of instances induce integer programs with less than one thousand binary decision variables, which are solved in a few seconds by the open-source branch-and-bound solver GLPK 4.24 (Makhorin 2007). But for larger instances (more than 10,000 binary decision variables), several dozens of minutes are needed to obtain near-optimal solutions, which is not acceptable for users. Thus, the challenge was to speed up this solution technique while avoiding the use of a commercial solver (which is of interest to lower development and maintenance costs). A main observation is that, in our case, larger integer programs are difficult to solve due to numerous symmetries. Indeed, when the number of available TV breaks in a day is large, audiences (and thus prices) do not vary much between two consecutive breaks, inducing many equivalent branches in the branch-and-bound tree. Since assigning commercials to consecutive TV-breaks is undesirable, a good preprocessing heuristic consists of restricting the set of available TV breaks (and thus the search space) by randomly removing one available TV-breaks among two consecutive ones. In the same way, additional soft constraints have been added in order to put lower and upper bounds on the number of commercials planned, which also makes sense from a business perspective. Ultimately, the resulting integer programs are all solved in less than 1 minute on a standard computer using GLPK 4.24 (Makhorin 2007). TF1 paid €70,000 to develop this OR engine. This software is used daily by the sales front office to build all the plans for the 15 theme channels managed by the group, generating a total revenue of €150 million per year.

4. Back-Office Software: Optimizing the Revenue of TF1

4.1. Business Context

The French TV advertising market is based on “sales openings”. This means that every other month, the main channels open their reservation planning for a period of 60 days (around 5,000 commercial breaks, each with a different price). Advertising companies then send their requests to their favorite channel. For instance, Dior might send the following request to TF1: “I want to buy 30 seconds in the 8:40 PM commercial break on May 24th for my Miss Dior perfume”. TF1 receives up to 50,000 such requests. One or two weeks later, TF1 returns its acceptance decision for each request. Then, each advertising spot can be cancelled or moved until a few days before its broadcasting date, new bookings can be made, late buying is possible under different conditions, etc. It should be noted that there is no auction mechanism as prices are fixed by TF1 (but can depend on the product sector for instance). Although the last stage of this process is a classical yield management situation similar to airline reservations or hotel bookings, the opening stage offers a unique opportunity to perform a global revenue optimization. Making the right decisions during this stage is crucial because 60 % to 70 % of airing time sold at this moment is eventually broadcasted without modification or cancellation (and the received requests generally exceed the total airing time for sale).

4.2. Optimization and Difficulty

For each commercial break, the problem can be seen as a multi-dimensional knapsack. We want to select a maximal price subset of requests for this commercial break. The sum of durations cannot exceed available duration; at most, one product is accepted per commercial sector (cars, perfume, toys, etc.) and 6 spots have small durations (less than 11 seconds). In essence, the economic function to be maximized is the total revenue, that is the sum of the prices of all accepted requests. But this objective ignores the importance of customer satisfaction (here customers are the advertising companies contracting with TF1). Indeed, even if refusing all requests from Coca-Cola while accepting all requests from Pepsi would yield optimal revenue for this period, an unhappy Coca-Cola could decide to severely decrease its requests for the next opening. Ensuring a certain acceptance rate for each customer is thus a priority (the acceptance rate is the price of accepted requests for this customer divided by the total price of its requests), meaning that the first euros accepted in each budget are more valuable than the last. In our case the objective is to achieve the best possible equity which means that increasing the least accepted budget by €1 is always preferable to increasing any other budget by €1. Technically speaking it means that the objective to be maximized is not directly the sum of the accepted budgets for each customer, but rather the sum of a concave "utility" function of each customer accepted budget. With this equity constraint the problem is not separable anymore: each knapsack (commercial break) cannot be solved independently from the others. In this context, a natural heuristic proceeds as follows:

1. Let C be the set of advertising companies;
2. **While** C is not empty **do**
   3. Select c in C with minimum acceptance rate;
   4. **If** c has no acceptable request anymore **then** remove c from C;
   5. **Else** accept one of the acceptable requests from c;

Naturally, the revenue obtained with this procedure is smaller than the sum of optimal revenues for each knapsack. If the request accepted in step 5 is picked at random, the gap is close to 4 %. It represents the cost of the equity constraint. It may seem a small impact but in fact the revenue of a two-month period amounts to hundreds of millions of euros. Now, we will show that this cost can be dramatically reduced by extracting sensitivity information from dynamic programs.

4.3. Solution Technique

The multi-dimensional knapsack for each commercial break can be modeled as a dynamic program. Within this model an optimal packing for a commercial break is an optimal path in the layered graph representing the dynamic program. An interesting byproduct of the computation of this optimal path is that it gives the value of the optimal path from each node of this graph to the terminal node (the so-called Bellman values). Now, if we perform this computation twice (from left to right and from right to left) we obtain for each node the value of the optimal path from the initial node and to the terminal node, namely the value of the optimal path crossing this node. Actually, considering all arcs associated to the acceptance of a certain request we can extract the value of the best packing containing this spot, and the value of the best packing excluding this spot (see appendix for details). The difference between the former and the latter is called the regret for this request, i.e. the loss induced by the refusal of this request. This regret is positive if and only if the request belongs to an optimal packing for this break and strictly positive if this optimum is unique.

Let us consider a 60-second commercial break for which we have four requests for 20, 20, 30, and 30 second durations, from different sectors, each paying the same price of €1000/second. The best packing with a 30-second spot is €30,000+€30,000 while without one of the 30-second spots, the best possible revenue is €30,000+€20,000, hence the regret for 30-second requests is €60,000−€50,000 or +€10,000. A similar reasoning for 20-second requests leads to a regret of −€10,000. Computing regrets for all requests merely requires solving each dynamic program twice and it makes it possible to replace the random selection in step 5 of our algorithm by the choice of the request with maximum regret among those of the selected company. Intuitively it means that rather than accepting a random spot we pick a spot fitting especially well with other requests for the same commercial break, or at least inducing the smallest possible loss on its break (when all regrets for the selected company are negative).

Of course, each time a spot is accepted, the regrets for requests on the same commercial break must be recomputed. Assume that we have a third 20-second request in our previous small example. In this case regrets are initially 0 for each of the five requests since each belongs to one of the two optimal packings: 30+30 or 20+20+20. If a 20-second spot is accepted for instance, the situation changes immediately. In the residual dynamic program the 30-second spots have a negative regret (−€10,000) while the 20-second spots have a positive regret (+€10,000). Finally, the number of dynamic programs to be computed cannot exceed the number of requests plus the number of commercial breaks. In practice, our Java implementation of this algorithm runs in less than three minutes. On average, we reach total revenue only 0.2 % smaller than would be obtained without taking customer satisfaction (equity) into account.

Beyond the optimization performance, we would like to emphasize the robustness and maintainability of this approach. Not only does our algorithm outperform a previous approach based on local search with ejection chains, it is also much easier to maintain thanks to its greedy behavior. Once a spot is accepted this decision is never reconsidered which means that if a user notices a questionable packing in a commercial break, the filling of this break can be easily traced and analyzed. The separation between a basic main loop and a sophisticated "oracle" (the dynamic programs providing sensitivity information) proved very useful as well. Indeed the problem presented here was simplified for the sake of readability. In the real application, the utility function is not strictly concave but becomes linear beyond a certain acceptance threshold, products can have multiple sectors, a sector may appear twice in a commercial break under certain conditions, some sectors have a priority during specific time slots, an additional price can be paid to get the first or last position of a commercial break, and so on. New offers and prices are also created every year or for special events like the World Cup. A large part of the general conditions of sale must be formalized as constraints in the model of the basic main loop so that for each break, the returned set of requests respects all priorities and compatibility constraints. On the contrary, the oracle can approximate or even omit some technical details without seriously affecting the final revenue. This simplicity and maintainability in an evolving context are representative of the qualities taken into account when choosing an algorithm for OR.

5. Lessons Learned

As illustrated by the above examples, our activity ranges from consulting studies to critical front-office software. Similar revenue optimization projects have been conducted for most aspects of the sales process including late buying orders, preferred positions dispatch, cancellation forecasts and yield management strategies. These systems allow TF1 to react better and more quickly to customers' requests than its competitors, thus reinforcing its leading position in the TV advertising market. They also contribute to optimizing the usage of its limited available air-time inventory.

The management of TF1 Advertising (*TF1 Publicité*) estimates the induced increase of revenue for TF1 at €20 million per year.

In this final section we try to present the principles that we successfully applied in this mission, as well as on other projects conducted in the Bouygues group or outside of it (inventory routing, construction site planning, road maintenance, contact center scheduling, and so on).

5.1. Modeling and Solving

From an algorithmic point of view, we try not to use a single, preferred technique, using instead mathematical programming (linear and integer programming, Lagrangian relaxation, etc.), constraint programming, local search, dynamic programming, or dedicated constructive algorithms depending on the problem. We also expect our staff to have good programming skills since we are convinced that materializing our analysis into operational software is an essential part of our job. The conjunction of these two requirements makes us a team of PhD-programmers. When specifying a problem we are more and more reluctant to add hard constraints to a model, except when expressing physical limitations, e.g. two messages cannot be simultaneously broadcasted by the same channel. More precisely, we try to make sure that finding a feasible solution is easy. Indeed, the worst situation for an OR system is when the end-user sees a "no solution found" message box. For instance, in our third example (sales opening) refusing all requests is a feasible solution, because the minimum acceptance rate for each advertiser is not modeled as a constraint but as a first-rank objective (minimization of the sum of distances to minimum rates). Even when a solution violating such "soft" constraints is unacceptable, seeing it is very useful for the user to detect the cause of the problem, which is almost always an inconsistency in input data.

5.2. Project Management

Presenting solutions to the client is the best way of identifying data inconsistencies or specification lapses. In order to address these issues as early as possible in the course of our projects, we tend to schedule an early beta delivery in the project planning. This beta version usually requires a few weeks of work and consists of an algorithm (generally greedy) returning a valid solution, possibly exceeding the expected execution time or omitting a few constraints. The goal of this release is twofold. First of all, it requires a set of input data from the client to verify that real data satisfies the preconditions stated in the specifications. Scheduling this task early is crucial because when inconsistencies are identified, fixing them can be time-consuming for the client and should be started as soon as possible. It is also a prerequisite for the development of the actual optimization algorithm. Incidentally, an efficient way to stress the importance of data in an OR project is to include in the contract this necessary delay between the delivery of valid datasets by the client and the delivery of the software by the vendor. In fact, it is arguably the only way. The second advantage of this beta release was illustrated in our "advertising plan" example. Presenting first solutions often leads to specifications being refined. It can also demonstrate the feasibility of the project and its potential return on investment. Indeed, we have seen cases where end-users doubted that an automatic system could produce useful solutions and thus repeatedly postponed change management tasks until confidence was restored by the first solutions presented.

5.3. Software Integration

We cannot emphasize enough the importance of data management and information technology in real-world OR applications. We have always had an IT expert on our team who can plug our algorithms into any information system. After 15 years of experience, we can say that this role is vital to our business. Even earlier than the beta version mentioned in the previous paragraph, we always deliver an empty shell or alpha version, making sure that communication between our algorithm and the host application works well. Usually it can be delivered a few days after the approval of detailed application programming interfaces. For most of our applications, we work in partnership with the IT department of our client. For instance, in the front-office application described in Section 3, user interfaces and interactions with the information system were developed by TF1 while we (the corporate lab) implemented the optimization algorithm. This one was developed in the C# 2.0 language, using GLPK as the native ISO C library. In this way, the integration of our optimization software into TF1's information system, based on Microsoft .NET framework, was simply done by delivering two Windows libraries (DLL). Indeed, our experience on various applications within the Bouygues group shows that if bridges between heterogeneous programming frameworks (for example, Java and .NET) are always possible, sticking to our customer's environment is nevertheless the safest choice.

5.4. Research

In conclusion, we would like to point out that our OR team acts both as a research center and as a profit center. Despite our corporate positioning, our subsidiaries pay for our services as if we were an external company. This mechanism ensures that we spend the right amount of time on the right project. Let us consider the three examples described in this paper. Internet sales are roughly ten times smaller than thematic channel revenues which are in turn ten times smaller than the income of our general interest channel (€1.4 billion). Not surprisingly, the costs of these three projects were consistent with these stakes.

On the other hand, around 30 % of our time is devoted to academic research. This research activity sometimes consists of scientific developments on operational projects, considering special cases, lower bounds or alternative solution approaches. The IT culture of the team also led us to develop optimization software like the constraint programming solver Choco (Laburthe 2000) or recently an innovative black-box local-search solver for 0-1 programming, namely LocalSolver (Benoist et al. 2011). Our scientists can also use this research time to attack academic problems. This presence in the academic community keeps our technical knowledge up to date or at least gives us an invaluable network of scholars to draw upon when facing a problem outside of our field of expertise. This is how we found the state-of-the-art UCB algorithm for our bandit problem in Section 2. This academic facet of our work also attracts high caliber candidates when opening a job or intern position. In practice, some of the optimization engines that we deliver to our customers are directly issued from this research activity. For instance, the ordering of ad messages within a commercial break is optimized with the above-mentioned LocalSolver. Even apparently useless research can turn out to be extremely profitable: the algorithm described in Section 4 was directly inspired by the technique developed by Benoist et al. (2001) for a sport scheduling problem.

Acknowledgments
We would like to thank all the people who have developed and maintained the partnership between TF1 Publicité and the Bouygues e-lab during these 15 years. These people include Éric Bourreau, Yves Caseau, Jacques Deregnaucourt, Étienne Gaudin, Emmanuel Guyot, François Laburthe, Hugues Laigneau, Bruno Martin, Jacques Masson, Benoît Rottembourg. We also express our gratitude to the anonymous referees for their meticulous reviewing, which has resulted in an improved paper.

Appendix. Sensitivity Analysis on Commercial Breaks

Consider a commercial break of duration D, for which K requests have been made whose durations and prices are $d_1, d_2, \ldots, d_K$ and $p_1, p_2, \ldots, p_K$ respectively. We assume here that all products have distinct commercial sectors and all messages have a duration larger than 11 seconds, that is to say that only the duration constraint applies. However the following reasoning remains valid for the dynamic program modeling all three constraints (duration, sectors and small sizes). Classically, we can define $\text{backward}[d,k]$ as the maximum possible revenue that can be obtained by selecting requests from 1 to k, for a total duration smaller than d. With this formalism the maximum revenue for this commercial break is $\text{backward}[D,K]$ and can be computed with the recursive formula $\text{backward}[d,k] = \max(\text{backward}[d,k-1], \text{backward}[d-d_k, k-1] + p_k)$. Symmetrically we can define $\text{forward}[d,k]$ as the maximum possible revenue that can be obtained by selecting requests from $k+1$ to K, for a total duration smaller than $D-d$, and we have $\text{forward}[d,k] =$

$\max(\text{forward}[d,k+1], \text{forward}[d+d_k,k+1]+p_k)$. Now the best possible possible revenue among all selections including request k is directly available as

$$\max_{d\in[0,D-d_k]}(\text{backward}[k-1,d]+p_k+\text{forward}[k+1,d+d_k])$$

while the best possible possible revenue among all selections *not* including request k is directly available as

$$\max_{d\in[0,D]}(\text{backward}[k-1,d]+\text{forward}[k+1,d])$$

Finally with only twice the complexity of a classical dynamic programming approach, we obtain sensitivity information for each request on this commercial break.

References

Auer, P., N. Cesa-Bianchi, P. Fisher. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* **47** 235–256.

Benoist, T., B. Estellon, F. Gardi, R. Megel, K. Nouioua. 2011. Localsolver 1.x: a black-box local-search solver for 0-1 programming. *4OR: A Quarterly Journal of Operations Research* **9**(3) 299–316. `http://www.localsolver.com`.

Benoist, T., F. Laburthe, B. Rottembourg. 2001. Lagrange relaxation and constraint programming collaborative schemes for travelling tournament problems. *Proceedings of the 3th International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2001)*. 15–26.

Bollapragada, S., H. Cheng, M. Phillips, M. Garbiras, M. Scholes, T. Gibbs, M. Humphreville. 2002. NBC's optimization systems increase revenues and productivity. *Interfaces* **32**(1) 47–60.

Gabszewicz, J., D. Laussel, N. Sonnac. 1999. TV-broadcasting competition and advertising. Tech. Rep. 99-72, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).

Goettler, R.L., R. Shachar. 2001. Spatial competition in the network television industry. *The RAND Journal of Economics* **32**(4) 624–656.

Ignizio, J. P. 1983. Generalized goal programming: an overview. *Computers & Operations Research* **10**(4) 277–289.

Kornbluth, J. 1973. A survey of goal programming. *Omega* **1**(2) 193–205.

Laburthe, F. 2000. Choco : implementing a CP kernel. *Proceedings of the Post-Conference Workshop of CP 2000 on Techniques for Implementing Constraint Programming Systems (TRICS 2000)*. 71–85. `http://choco.emn.fr`.

Makhorin, A. 2007. GLPK: GNU Linear Programming Kit, version 4.24. `http://www.gnu.org/software/glpk/`.

Nakamura, A., N. Abe. 2005. Improvements to the linear programming based scheduling of web advertisements. *Electronic Commerce Research* **5** 75–98.

Pandey, S., C. Olston. 2007. Handling advertisements of unknown quality in search advertising. B. Schölkopf, J. Platt, T. Hoffman, eds., *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 1065–1072.