

Programmation de campagnes publicitaires sur les chaînes thématiques du groupe TF1

Frédéric GARDI et Thierry BENOIST

Bouygues SA / DGITN / e-lab, Paris
{fgardi,tbenoist}@bouygues.com

1 Introduction

TF1 Publicité, régie publicitaire du groupe TF1, gère un portefeuille de chaînes thématiques. Afin de faciliter la vente de spots publicitaires sur ces chaînes, la régie propose aux annonceurs publicitaires d'acheter les spots par pack. Grossièrement, un pack permet de consommer un budget tout en garantissant à l'annonceur une certaine audience sur l'ensemble de sa campagne publicitaire. La particularité des nouvelles offres commerciales, qui entreront en vigueur début 2008, est que le nombre de spots vendus à travers un pack n'est plus fixe, mais est fonction de l'audience que l'on souhaite atteindre. Les packs sont programmés en temps réel à la volée, c'est-à-dire que les premiers arrivés sont les premiers servis, jusqu'à ce que le planning soit saturé (ou qu'il n'y ait plus de demande).

Programmer manuellement des packs est une tâche longue et fastidieuse, d'autant que derrière cette programmation se cache une problématique d'optimisation. En effet, pour le programmeur, le but est de capter l'intégralité du budget d'un annonceur en satisfaisant au mieux un ensemble de contraintes commerciales qui dépasse souvent le cadre contractuel (par exemple assurer une répartition homogène des spots sur la campagne de l'annonceur). Afin d'aider le programmeur dans cette tâche, nous avons développé une solution logicielle permettant la programmation de packs publicitaires de façon automatique et optimisée. Toutefois, précisons que notre logiciel se veut avant tout un outil d'*aide* à la décision.

Dans cette note, nous présentons dans ses grandes lignes la solution par *goal programming* [1] que nous avons implémentée pour résoudre de manière effective le problème, qui s'apparente à un problème de sac-à-dos à dimensions et objectifs multiples. Le lecteur intéressé pourra également consulter le papier de Benoist *et al.* [2] traitant de la problématique de programmation des packs relative aux anciennes offres commerciales.

2 Présentation du problème

Sur toute chaîne télévisuelle, la grille des programmes fait apparaître sur une semaine type un ensemble de plages de durée donnée, appelées écrans, dédiées au placement de spots publicitaires. Les enquêtes et études marketing menées par la société Médiamétrie, spécialiste français de la mesure scientifique de l'audience, fournissent une estimation fiable de l'audience générée par chaque écran de la grille, sur chaque cible marketing (par exemple les catégories socioprofessionnelles supérieures); l'audience est mesurée en GRP (de l'anglais *gross rating point*), 1 GRP correspondant à 1 % de la cible considérée. Une fois tarifé (nous ne détaillerons pas ici le processus de tarification), l'écran possède un prix et une audience par cible. Ainsi, le prix d'un spot est calculé comme le prix de l'écran dans lequel il est placé, multiplié par un indice fonction de la durée du spot (appelée format), pénalisant d'autant plus les formats qu'ils s'écartent de la durée de référence 30 secondes.

L'annonceur dispose d'un budget pour réaliser une campagne publicitaire sur une chaîne thématique. Une campagne s'étale sur une période pouvant aller de quelques jours à plusieurs semaines. La régie met à la disposition de l'annonceur une gamme de packs. Un pack garantit à l'annonceur une certaine audience sur la cible souhaitée, sur la totalité de sa campagne; plus exactement, le pack assure un certain coût du GRP, rapport entre la somme des prix des spots (en valeur 30 secondes) sur la somme des GRP générés. Une seconde garantie apportée par le pack est la répartition des GRP dans chacune des grandes zones de la grille (matin, déjeuner, *prime time*, week-end, *etc.*).

Une fois toutes ces données en main, le programmeur doit construire un "chemin publicitaire" sur l'étendue de la campagne, c'est-à-dire placer un ensemble de spots dans les écrans de la grille

répondant aux besoins de l'annonceur (un spot peut être placé dans un écran dès lors que son format est inférieur à la durée résiduelle de l'écran). La difficulté est alors de capter l'intégralité du budget de ce dernier tout en satisfaisant les contraintes définies par le pack choisi, ainsi qu'un ensemble non négligeable de contraintes dites "métier". Certaines de ces contraintes sont nécessaires : assurer une homogénéité verticale (même nombre de spots chaque jour) et horizontale (même nombre de spots dans chaque zone), assurer une certaine répartition verticale (écartement des spots en journée) et horizontale (éviter deux spots à la même heure deux jours consécutifs). D'autres sont laissées à la guise de l'utilisateur : exercer une pression horizontale (placer 25 % des spots la semaine de lancement d'un nouveau produit) ou verticale (placer un nombre minimum de spots dans certaines tranches de la grille, par exemple autour de l'émission Le Grand Jury sur LCI ou CauetTiVi sur TF6), mélanger des formats et spécifier leur répartition (par exemple, 30 % des GRP de la campagne en format 10 secondes et 70 % en format 20 secondes).

Lorsqu'un écran est presque saturé (durée résiduelle inférieure à 30 secondes), il est préférable de faire en sorte que sa durée résiduelle corresponde à un format de spot demandé par les annonceurs, de manière à pouvoir saturer l'écran par la suite (tout résidu finalement invendu n'étant que pure perte). La probabilité de demande des différents formats étant connue, l'idée est d'ajouter comme objectif annexe un terme permettant de minimiser la perte potentielle engendrée par ces durées résiduelles (cet objectif permet en outre d'assurer une consommation équilibrée des durées d'écrans).

3 Une solution par *goal programming*

La difficulté du problème se pose à deux niveaux. Tout d'abord, son caractère NP-difficile (sac-à-doc à dimensions multiples) rend difficile une approche exacte en temps réel, d'autant plus lorsque le nombre d'écrans de la grille est grand et la campagne à planifier est longue. Ensuite, l'algorithme doit fournir une solution au problème posé, quelque soit les contraintes posées par le programmeur (en particulier lorsque la grille commence à être saturée). En fait, toutes les contraintes du problème sont molles : si une contrainte ne peut être satisfaite, il faut pouvoir la dégrader. Par exemple, ce sera le cas lorsque la grille commence à être saturée ou bien lorsque des contraintes entrent en contradiction les unes avec les autres (par exemple, une forte pression verticale empêchera la répartition des GRP sur les zones telle que promise par le pack). Ce besoin de souplesse apparaît comme déterminant dans la problématique de programmation, puisqu'il va de pair avec la volonté de faire la meilleure offre commerciale possible au client.

Notre schéma de résolution est basé sur une approche exacte par programmation linéaire en variables mixtes. La difficulté réside avant tout dans la modélisation globale du problème par *goal programming* (GP) nécessitant toute la panoplie des techniques de type GP : lexicographie, pondération, intervalle de satisfaction, min-max [1]. Afin d'accélérer la résolution, le *branch-and-bound* est guidé par l'addition de contraintes molles cadrant de près le nombre spots du chemin recherché (ces bornes, obtenues de façon analytique, correspondent également à une réalité métier).

Notre solution logicielle, développée en C# et basée sur le solveur libre GLPK 4.13 [3], obtient des solutions quasi optimales en 30 secondes sur un ordinateur équipé d'un processeur Intel Xeon 3 GHz. Toutefois, un temps d'exécution plus long s'avère parfois nécessaire à l'obtention d'une première solution, en particulier sur la grande grille de LCI (qui contient environ 100 écrans par jour). Nos travaux futurs vont donc porter sur la réduction du temps d'exécution, que nous pensons possible par élimination de certaines symétries.

Références

1. B. Aouni, A. Hassaine, J.-M. Martel (2006). Les préférences du décideur dans le goal programming : état de l'art et perspectives futures. In *MOSIM 2006, la 6ème Conférence Francophone de Modélisation et Simulation*. Rabat, Maroc.
2. T. Benoist, E. Bourreau, B. Rottembourg (2007). The TV-break packing problem. *European Journal of Operational Research* 176, pp. 1371–1386.
3. A. Makhorin (2006). GLPK : GNU Linear Programming Kit, version 4.13. <http://www.gnu.org/software/glpk/>