



HAL
open science

Affichage de publicités sur des portails web

Victor Gabillon, Jérémie Mary, Philippe Preux

► **To cite this version:**

Victor Gabillon, Jérémie Mary, Philippe Preux. Affichage de publicités sur des portails web. Extraction, Gestion des Connaissances (EGC), Jan 2010, Tunisie. pp.110-120. hal-00772531

HAL Id: hal-00772531

<https://hal.science/hal-00772531v1>

Submitted on 10 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Affichage de publicités sur des portails *web*

Victor Gabillon^{*,**} Jérémie Mary^{*,**} Philippe Preux^{*,**}

*Équipe-Projet SequeL

INRIA CR-LNE, 40 Av. Halley, 59650 Villeneuve d'Ascq, France

prenom.nom@inria.fr, <http://sequel.futurs.inria.fr/>

**Laboratoire d'Informatique Fondamentale de Lille (LIFL, UMR CNRS Université de Lille 1 & 3),
Cité Scientifique, 59655 Villeneuve d'Ascq Cedex, France

Résumé. Nous nous intéressons au problème de l'affichage de publicités sur le *web*. De plus en plus d'annonceurs souhaitent maintenant payer uniquement lorsque quelqu'un clique sur leurs publicités. Dans ce modèle, l'opérateur du portail a intérêt à identifier les publicités les plus cliquées, selon ses catégories de visiteurs. Comme les probabilités de clic sont inconnues *a priori*, il s'agit d'un dilemme exploration/exploitation. Ce problème a souvent été traité en ne tenant pas compte de contraintes provenant du monde réel : les campagnes de publicités ont une durée de vie et possèdent un nombre de clics à assurer et ne pas dépasser. Pour cela, nous introduisons une approche hybride (MAB+LP) entre la programmation linéaire et les bandits. Nos algorithmes sont testés sur des modèles créés avec un important acteur du *web* commercial. Ces expériences montrent que ces approches atteignent une performance très proche de l'optimum et mettent en évidence des aspects clés du problème.

1 Introduction

La publicité sur Internet est une source de revenus considérables. Une migration est en cours d'un modèle économique « à l'impression » (CPM) à un modèle « au clic » (CPC), *i.e.*, l'annonceur paye quand l'internaute clique sur sa publicité (CPC), et non seulement quand celle-ci est affichée (CPM) sur une page *web*, dans un certain contexte (compte tenu de la page, caractéristiques de l'internaute, ...). C'est un changement majeur car dans le CPM, le risque économique est supporté par l'annonceur, alors que dans le CPC, c'est l'opérateur du portail *web* où sont affichées les publicités qui prend les risques. Dans le CPC, cet opérateur doit optimiser l'affichage des publicités afin de maximiser la quantité de clics sur celles-ci. De plus, comme l'annonceur ne court aucun risque, le stock de publicités à afficher est bien plus grand que l'espace disponible sur les pages *web* et donc, l'opérateur du portail doit sélectionner judicieusement les publicités qu'il affiche pour réaliser au mieux ses contrats avec les annonceurs. L'aptitude à afficher des publicités sur lesquelles les internautes cliqueront effectivement est donc une question critique. Même une amélioration faible du taux de clics peut se révéler cruciale du fait de la masse énorme de publicités affichées chaque jour sur un portail important et que cette amélioration est obtenue avec un coût très faible. Aussi, c'est

sans surprise que l'on constate l'intérêt d'acteurs majeurs d'Internet comme Yahoo ! et Google pour ces questions (Mehta et al., 2005; Pandey et Olston, 2006; Chakrabarti et al., 2008). Par ailleurs, ce problème possède des contraintes pratiques qui sont rarement prises en compte dans les articles de recherche du domaine :

- durée de vie : l'annonceur spécifie une durée de vie pour sa publicité (correspondant à une période de soldes par exemple). De plus, l'annonceur souhaite que ses publicités soient présentées avec une fréquence relativement uniforme durant cette période ;
- budget de clics fixé : l'annonceur achète un certain nombre clics et n'en paiera pas plus ; par contre, si le nombre de clics obtenus pendant la durée de vie de sa publicité est inférieur à ce budget, il paiera finalement beaucoup moins l'opérateur du portail.

Dans ce papier, nous ne nous intéressons qu'au modèle CPC. Nous souhaitons concevoir des algorithmes efficaces, effectivement capables de sélectionner en temps réel les publicités à afficher, de la meilleure manière possible. Une originalité majeure de notre article vient du fait que nous nous appuyons sur une modélisation aussi réaliste que possible du problème effectivement rencontré par les opérateurs de portail *web*. Dans la suite de cet article, nous commençons par définir le problème à résoudre dans des termes plus précis. Nous nous livrons alors à une revue de la littérature dans la section 3. Ensuite, dans la section 4, nous décrivons notre approche qui est un processus itératif qui hybride la résolution d'un programme linéaire, avec un algorithme à base de bandits multi-bras. La section 5 fournit quelques résultats expérimentaux saillants, puis nous concluons.

2 Le problème et l'idée de notre solution

2.1 Définition du problème

Le problème que nous étudions est séquentiel dans le temps. À un instant t , nous disposons d'un ensemble de K^t campagnes publicitaires en cours $Ad_1, Ad_2, \dots, Ad_{K^t}$. À l'instant t , chaque campagne Ad_j dispose d'un budget restant B_j^t qui est le nombre de clics que cette publicité doit encore recevoir pendant sa durée de vie restante L_j^t . B_j^t et L_j^t sont connus : ils sont initialement spécifiés dans le contrat liant l'annonceur et l'opérateur du portail *web*. Les visites au portail *web* sont supposées segmentées en N profils notés *Profile_i*, $i \in 1, \dots, N$. À chaque profil est associée sa probabilité quotidienne de visite sur le portail *web* R_1, \dots, R_N ($\sum_1^N R_i = 1$). La page (url) requêtée fait partie du profil de telle manière qu'une même personne requêtant deux pages différentes est généralement associée à deux profils différents. On suppose également que ces probabilités R_i sont connues : elles sont facilement estimées à partir des fichiers *logs* du site *web* : le nombre de visites est tellement grand que la confiance sur ces estimations est grande. Enfin, outre le fait que les campagnes de publicités ont une durée de vie finie et fixée à l'avance, de nouvelles campagnes apparaissent quotidiennement. Ceci étant posé, quand un internaute requête une page, on suppose qu'une publicité est sélectionnée pour être affichée sur la page. L'objectif est donc que pendant sa durée de vie, chaque publicité soit cliquée en fonction de son budget. Pour simplifier la présentation, nous supposons que toutes les campagnes de publicités apportent le même bénéfice financier en cas de succès, et ont le même coût en cas d'échec. De plus, les annonceurs souhaitent que leurs publicités soient cliquées à un rythme à peu près uniforme tout au long de leur durée de vie.

2.2 Présentation informelle de notre approche

Afin de maximiser le nombre total de clics, une information importante est le taux de clic (*Click-Through Rate* ou CTR en anglais) d'un certain profil, sur une certaine publicité. Notons $p_{i,j}$ la probabilité qu'un visiteur du profil $Profile_i$ clique sur la publicité Ad_j . Ces probabilités $p_{i,j}$ sont inconnues. Une bonne estimation de ces probabilités permet d'afficher des publicités qui ont une plus grande probabilité d'être cliquées individuellement par chacun des visiteurs du portail. Cependant, cet apprentissage étant réalisé en ligne, le principal problème consiste à combiner l'estimation de ces paramètres inconnus avec l'exploitation de leur estimation courante. Ce problème peut-être formulé dans le cadre des bandits multi-bras, les publicités jouant le rôle de bras. Plusieurs solutions documentées dans la littérature sont des exemples de formulation sous forme de bandits (Kakade et al. (2008), Langford et Zhang (2008)). Néanmoins, l'existence de budgets de clics finis, accompagnée à la durée de vie des campagnes publicitaires assez courte, nécessite des approches algorithmiques fournissant des solutions de bonnes qualités dans des temps courts, autrement dit, non asymptotiquement. En effet, on ne peut pas se contenter ici d'algorithmes prouvés optimaux avec des temps de convergence dépassant la durée de vie des campagnes, ou un nombre de visites d'internautes infini, un nombre de clic infini, ... Le fait que les budgets de clics soient finis rend le problème combinatoire, dans un contexte incertain, stochastique et évoluant dans le temps. L'aspect fini crée des dépendances entre les visites et les publicités qui sont présentées aux différents internautes et aux différents profils d'internautes. Ainsi, simplement afficher la publicité qui a la plus grande probabilité de clic à un profil donné n'est pas la stratégie optimale. Une technique envisageable pour déterminer la stratégie de sélection des publicités est la programmation linéaire (PL) basée sur l'estimation courante des probabilités de clics de chaque profil sur chaque publicité $p_{i,j}$. Néanmoins, comme la PL nécessite que lui soient spécifiés tous les paramètres du problème (nombre de visiteurs par profils, probabilités de clics, ...), elle nécessite aussi que le problème ne change pas au cours du temps. Ces paramètres du problème sont inconnus et doivent être estimés, cette estimation devant être mise constamment à jour. Aussi, nous proposons de mixer la programmation linéaire aux bandits pour résoudre ce problème dans un contexte incertain, stochastique et évoluant au cours du temps.

3 Travaux connexes

Nous brossons un rapide tour d'horizon des travaux existants sur ce problème de sélection de publicités pour les pages *web*. Nous découpons cette présentation en deux parties, la première concernant les bandits, la seconde concernant la programmation linéaire.

3.1 Bandits multi-bras

La sélection d'une publicité pour un visiteur peut se formuler simplement comme le choix d'une machine à sou, un bandit manchot, dans le cadre des bandits multi-bras (MAB). Depuis le travail initial de Robbins (1952), ce problème de l'allocation optimale des tirages de bras parmi un ensemble de bras a vécu une révolution avec l'introduction de l'algorithme UCB par Auer et al. (2002). Face à un ensemble de bras, chaque bras ayant une probabilité de succès fixe et inconnue, l'objectif est de minimiser le nombre de tirage de bras sous-optimaux ; l'idée

est alors de tirer chacun des bras et, en fonction des succès observés, estimer les probabilités de succès sur chacun des bras, et tendre vers le choix exclusif du bras ayant la probabilité de succès maximale. À côté de ce cadre très épuré, de nombreuses variations ont été proposées et étudiées dans différents contextes. Les « bandits contextuels » sont une telle extension dans laquelle une information est présente associée à chaque bras qui, on l'espère, doit aider au choix du bras optimal (Pandey et al. (2007), Langford et Zhang (2008), Wang et al. (2005), Kakade et al. (2008)). Dans le problème qui nous intéresse ici, ces informations seraient typiquement l'internaute, ou du moins des caractéristiques connues de l'internaute, la page requêtée, la date, l'heure, . . . Parmi ces travaux, Pandey et al. (2007) considèrent une segmentation des pages *web* et des publicités et bâtissent un bandit à deux étages : sélectionner d'abord le bon segment, puis sélectionner la bonne publicité pour ce segment. Dans Kakade et al. (2008), le contexte est un vecteur $x \in \mathbb{R}^d$. Inspiré par le perceptron, leur « banditron » est un algorithme dont l'objectif est de classer ces vecteurs sur K étiquettes, *i.e.* un bandit à K bras. Ici, le problème de sélection de publicités est réduit à un problème de classification multi-classes, avec un retour des bandits. Chakrabarti et al. (2008) analysent une version du MAB dans laquelle les publicités ont une durée de vie et un budget connus, voire qui sont stochastiques. Leurs résultats sont basés sur l'utilisation d'une connaissance *a priori* sur la probabilité de succès des bras et s'appuient sur cette dernière pour obtenir plus rapidement un bon bras à sélectionner. Pandey et Olston (2006) ont proposé une adaptation du cadre MAB dans le cas où des budgets de clics sont fixés aux publicités. Ces approches pourraient nous être utiles si elles permettaient d'induire les probabilités de clics. Mais, pour la plupart, elles ne font que sélectionner la « meilleure » publicité pour un certain contexte. Cependant, pour être optimal, l'allocation des publicités aux différents segments de profils doit être effectuée en tenant compte de ressources finies. Cette allocation peut être calculée par PL. De plus, ces travaux ne considèrent pas dans un même cadre unifié à la fois les limites de budgets, la création et la durée de vie limitée des campagnes publicitaires.

3.2 Programmation linéaire

Abe et Nakamura (1999) ont mixé un cadre MAB avec la programmation linéaire pour le problème de publicités en ligne, avec des contraintes sur les proportions d'impression. Néanmoins, il s'agit donc du modèle CPM et les auteurs n'ont considéré que des publicités ayant des budgets illimités, dans un contexte statique de campagnes publicitaires. Mehta et al. (2005) ont traité ce problème comme un problème d'appariement biparties, avec des contraintes de budgets limités. Cependant, ils ont supposé n'avoir aucune information *a priori* sur les visites des différents profils, alors qu'en pratique, nous avons une estimation de cette information. Mahdian et Nazerzadeh (2007) ont utilisé ces estimations tout en conservant une certaine quantité d'exploration pour pallier les incertitudes autour de ces estimations. L'extension à des budgets de clics finis est discutée dans le cas où les probabilités de clic sont connues et non apprises. De plus, ils s'intéressent à la maximisation quotidienne du revenu, ce qui n'est pas équivalent à avec une maximisation globale.

4 Combiner bandits et programmation linéaire

Nous désirons résoudre un problème d'optimisation combinatoire afin de déterminer les probabilités d'allocation de chaque profil pour chaque publicité. Cependant, l'élément principal, les probabilités de clic est inconnu. Elles doivent être estimées en ligne. Pour bien appréhender la suite, il est crucial de bien comprendre que les probabilités d'allocation et les probabilités de clics sont deux choses différentes : si les probabilités de clics étaient connues exactement, ce serait la même chose et la stratégie optimale serait une pure exploitation des probabilités de clics, en choisissant la publicité de manière gloutonne ; ici, les probabilités de clics sont estimées et cette estimation comprend donc des incertitudes. Aussi, face à un profil de visiteur donné, la publicité estimée comme étant celle ayant la plus grande probabilité de clic n'est peut-être pas celle qui a vraiment la plus grande probabilité de clic. Pour lever ce problème, il faut donc explorer, c'est-à-dire, proposer aussi des publicités pour lesquelles les estimations de probabilité de clic n'est pas la plus forte. Il faut donc judicieusement mixer exploration et exploitation pour pouvoir améliorer la stratégie courante. Ce double problème demande donc la combinaison d'une méthode d'optimisation combinatoire avec un algorithme d'apprentissage. La programmation linéaire et les bandits multi-bras seront les deux outils utilisés simultanément à cet effet : MAB estime les probabilités de clic et PL fournit une solution optimale de planification ; notons bien qu'ici, l'expression « optimale » est trompeuse car la solution calculée est la meilleure relativement aux estimations plus ou moins précises des probabilités de clic. Pour surmonter ces difficultés, une solution est d'adapter continûment la solution du problème aux nouvelles conditions observées. Ceci est réalisé en itérant simplement le processus d'estimation des probabilités, et en résolvant à chaque instant un problème d'optimisation combinatoire avec les estimations courantes sur les publicités actuelles.

4.1 Notre algorithme hybride : MAB+LP

Définissons à l'aide des notations introduites dans la section 2 le programme linéaire à résoudre. Notons H l'horizon, c'est-à-dire la prévision du nombre de requêtes de publicités (visites), paramètre dont nous préciserons le rôle par la suite. Solution du PL, chaque $x_{i,j}$ est le nombre de visites de profils i à allouer à la publicité j . Étant donné $(N, K^t) \in \mathbb{N}^2$, respectivement le nombre de profils et le nombre actuel de campagnes publicitaires, $p \in [0, 1]^{N \times K^t}$, les probabilités de clic, $(B^t, L^t) \in \mathbb{N}^{K^t} \times \mathbb{N}^{K^t}$ les budgets, et $R \in [0, 1]^N$, les proportions d'apparition des différents profils, PL calcule $x^t \in \mathbb{R}^{N \times K^t}$, la politique d'allocation solution du programme linéaire indiqué dans la table 1.

On suppose que les publicités sont préalablement triées par ordre croissant de durée d'affichage restante. Le programme linéaire contient $N \times K^t$ variables et $N \times K^t + N + K^t$ contraintes. L'objectif (eq. (1)) est de maximiser le nombre de clics. Les inégalités (2) expriment les contraintes sur le budget en clics des publicités. Les inégalités (3) expriment les contraintes sur le nombre de visites provenant de chaque profil. Les inégalités (4) expriment les contraintes sur la durée de chaque campagne publicitaire. Pour ces dernières inégalités, l'idée est de contraindre, pour chaque profil, l'allocation sur une publicité à être inférieure au nombre total de visites qui se dérouleront durant son existence. De plus, récursivement, l'allocation pour une publicité A doit aussi prendre en compte le nombre de visites déjà allouées aux publicités qui disparaîtront avant elle. Remarquons que L_j peut être considéré comme un

Affichage de publicités sur des portails web

$$\begin{aligned}
 \text{Maximiser} \quad & \sum_{\substack{1 \leq i \leq N \\ 1 \leq j \leq K^t}} x_{i,j} p_{i,j} & (1) \\
 \text{Sous contraintes} \quad & \sum_{1 \leq i \leq N} x_{i,j} p_{i,j} \leq B_j^t \quad \forall j \in \{1, \dots, K^t\} & (2) \\
 & \sum_{1 \leq j \leq K^t} x_{i,j} \leq R_i * H \quad \forall i \in \{1, \dots, N\} & (3) \\
 & \sum_{1 \leq k \leq j} x_{i,k} \leq L_j^t * R_i \quad \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, K^t\} & (4) \\
 & x_{i,j} \geq 0 \quad \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, K^t\} & (5)
 \end{aligned}$$

TAB. 1 – Programme linéaire à résoudre.

nombre de requêtes de publicités, car nous supposons que le nombre de visites est le même chaque jour (ce qui est faux mais qui n'affecte pas outre mesure notre solution qui essaye de maximiser le nombre de clics globalement et non jour après jour). Si les probabilités de clic $p_{i,j}$ étaient connues, la planification représentée par x serait, en espérance, la meilleure politique non-adaptative à suivre. Cependant, les CTRs doivent être appris en ligne. Une idée simple, déjà appliquée dans (Abe et Nakamura, 1999), est d'utiliser l'estimation courante de p pour générer la planification. Par exemple, après avoir observé un certain nombre de visites, un estimateur naturel des $p_{i,j}$ est $\hat{p}_{i,j}^t = \frac{NC_{i,j}^t}{NI_{i,j}^t}$ avec, pour chaque profil i et publicité j , $NC_{i,j}^t$ le nombre de clics observés jusqu'à l'instant t et $NI_{i,j}^t$ le nombre d'affichages réalisés jusqu'à l'instant t . Ainsi, à chaque nouvelle visite, $\hat{p}_{i,j}^t$ peut être mis à jour car il existe un couple (i, j) pour lequel $NI_{i,j}$ a été incrémenté (et peut-être son $NC_{i,j}$ associé). Dès lors, cette nouvelle estimation peut être prise en compte pour recalculer une politique d'allocation associée. L'algorithme présenté dans la table 2 découle de ces observations. En pratique, résoudre le PL après chaque requête d'une page web n'est pas raisonnable lorsque l'on doit gérer des millions de connexions par jour. Donc, pour rendre notre méthode réalisable et efficace, nous ne replanifions que toutes les T visites. Une analyse de sensibilité nous a permis de choisir T adéquatement. De plus, il s'agit d'un algorithme d'apprentissage en ligne qui doit estimer les $p_{i,j}$ utilisés pour le calcul de la solution du PL, à partir desquels les probabilités d'affichage des publicités pour les différents profils sont déduites (les $s_{i,j}^t$). De ce fait, nous devons trouver un équilibre entre l'exploitation de nos estimateurs courants et l'exploration de l'ensemble des actions possibles (afficher une publicité actuellement considérée comme ayant moins de chance d'être cliquée) d'améliorer nos estimations. Ce compromis « exploration / exploitation » est naturellement traité par le formalisme des bandits multi-bras. En se basant sur le cadre MAB, différentes manières d'introduire l'exploration dans la politique d'allocation sont envisageables, parmi lesquelles nous différencions deux familles :

Dévier de la planification du PL : les méthodes ε -gloutonnes peuvent être appliquées. Ceci implique de suivre la solution du PL avec (grande) probabilité ε et avec probabilité $1 - \varepsilon$,

Itération au temps t :	
Politique d'allocation :	
Estimateurs courants :	$\hat{p}_{i,j}^t$
Résoudre le programme linéaire :	$x^t = PL(\hat{p}^t, B^t, L^t, R, H)$
Calculer les probabilités d'allocation :	$s_{i,j}^t = \frac{x_{i,j}^t}{\ x_i^t\ } \forall i \in \{1 \dots N\}, \forall j \in \{1 \dots K^t\}$ avec $\ x_i^t\ = \sum_{j=1}^{K^t} x_{i,j}^t$
Visite :	Pour notre t^e visite, un $Profile_i$ apparaît.
Affichage :	Une publicité Ad_l est choisie selon la densité $s_{i,\cdot}^t$.
Clic :	Un clic se produit ou non.
Mise à jour :	$\hat{p}_{i,l}^t$ est mis à jour. $L_k^{t+1} \leftarrow L_k^t - 1 \quad \forall k \in \{1, \dots, K^t\}$ $B_l^{t+1} \leftarrow B_l^t - 1$ si un clic s'est produit. Une nouvelle publicité apparaît avec probabilité u (Mettre alors à jour K^t).

TAB. 2 – Principe de l'algorithme MAB+LP.

choisir une publicité aléatoirement. Notons cette stratégie d'exploration « PL- ϵ ».

Modifier le PL : Les $\hat{p}_{i,j}^t$ peuvent être modifiés pour que la planification inclut une part d'exploration. Pour ce faire, Abe et Nakamura (1999) ont utilisé les indices de Gittins. Il est aussi possible de remplacer les $\hat{p}_{i,j}^t$ par les valeurs UCB, ou par une réalisation aléatoire tirée selon la distribution *a posteriori* Bêta sur les probabilités moyennes de clic (voir Granmo (2008)).

4.2 L'horizon

Dans la réalité, le temps n'est pas borné. Aussi, pour résoudre le PL faut-il fixer un horizon de planification H^t qui peut varier au cours du temps et est décrémenté après chaque requête. Le trafic sur le site est continu et de nouvelles publicités apparaissent constamment tandis que d'autres disparaissent. Cet environnement dynamique influence la manière optimale de gérer les publicités. C'est pourquoi nous désirons maximiser le nombre total de clics, plutôt que de maximiser jour après jour le nombre de clics. Dès lors, les performances des politiques sont dépendantes de la manière dont les nouvelles campagnes sont créées ainsi que de leur qualité (attirent-elles l'attention des visiteurs ?). À ce stade, une question importante se pose : devons nous être gourmand avec nos publicités actuelles et distribuer à chaque visiteurs sa campagne supposée préférée car bientôt d'autres campagnes attractives seront générées ? Devons-nous plutôt nous montrer plus prévoyant et allouer les publicités en envisageant une pénurie de nouvelles publicités ? Régler le paramètre H revient à moduler entre une politique gourmande et une stratégie prévoyante pour les publicités actuelles. En effet, si nous allouons les prochaines visites comme si le nombre total de visites était petit, les budgets des publicités ne nous empêcheraient pas de proposer à chacun sa publicité préférée et l'algorithme jouerait de façon gourmande. Au contraire, en prévoyant une arrivée massive de visites à traiter, MAB+LP opérerait pour une politique plus prévoyante pour gérer les publicités actuelles. La question est de

Affichage de publicités sur des portails web

		Budget	100	100				Budget	100	100
H	R_i		Ad 1	Ad 2		H	R_i		Ad 1	Ad 2
20	1/2 →	Profile 1	10	0		300	1/2 →	Profile 1	125	25
	1/2 →	Profile 2	10	0			1/2 →	Profile 2	0	150
Planification avec $H = 20 \rightarrow$ Gourmand						Planification avec $H = 300 \rightarrow$ Prévoyant				
Profile 1 : 100% sur Ad_1						Profile 1 : 73% sur Ad_1 , 17% sur Ad_2				
Profile 2 : 100% sur Ad_1						Profile 2 : 100% sur Ad_2				

TAB. 3 – Effet du paramètre horizon H sur l'allocation des profils sur les publicités.

déterminer la valeur de H optimale en fonction de la manière dont sont créées les nouvelles campagnes publicitaires. Pour illustrer ces observations, considérons le problème ci-dessous où les valeurs $p_{i,j}$ représentées dans des tableaux possèdent des ordres de grandeur irréalistes par souci de clarté (les véritables CTR sont 10^3 fois plus petits) :

	Ad 1	Ad 2
Profile 1	0,8	0,1
Profile 2	0,8	0,5

Supposons que nous ayons deux campagnes publicitaires disposant chacune d'un budget de 100 clics, et qu'il existe deux profils de visites. Dans la table 4.2, deux politiques distinctes d'allocation calculées par le PL sont représentées pour deux valeurs de H : $H = 20$, et $H = 300$. Les deux tableaux contiennent les nombres de visites allouées à chaque couple

(Profile, Publicité) (les $x_{i,j}$). On retrouve le nombre de clics pour chaque couple en multipliant ces valeurs par leurs $p_{i,j}$ respectifs. Pour $H = 20$, l'allocation est gourmande alors que pour $H = 300$, l'algorithme prend en compte les budgets limités (par rapport à la grande demande liée au H grand) afin de bien répartir les ressources sur les profils. Cependant le réglage de H dépend fortement de la manière dont naissent les campagnes. Et comme le choix de H mène à des politiques distinctes, ce choix s'avère crucial et sera étudié expérimentalement dans la section suivante. Notons qu' $H = 1$ correspond exactement à une stratégie consistant à systématiquement afficher sa publicité préférée à chaque visiteur.

5 Expérimentations

Nous présentons dans cette partie des expériences montrant l'intérêt de notre approche. Nous nous intéresserons à deux aspects : l'écart de diverses approches avec la solution optimale et l'importance de l'horizon H sur la performance de MAB+LP. Nos expérimentations sont basées sur un modèle jouet conçu avec des experts d'Orange Labs avec le souci de reproduire les difficultés du problème réel. Chaque publicité possède ses caractéristiques (secrètes) qui font qu'elle est plus ou moins adaptée aux profils.

5.1 Mise en place

Dans la suite, les $p_{i,j}$ ont une probabilité de base de 10^{-4} , chaque fois qu'une caractéristique (conservée secrète) des publicités correspond à un élément de profil, la probabilité est

multipliée par 4. Dans les simulations, la moyenne de K^t est 30 ; il y a 54 profils de visites ; nous supposons que la distribution des visites est uniforme au cours de la journée. Le lecteur intéressé trouvera des détails supplémentaires pour reproduire les expériences dans Gabillon (2009).

5.2 Sans arrivée de nouvelles publicités

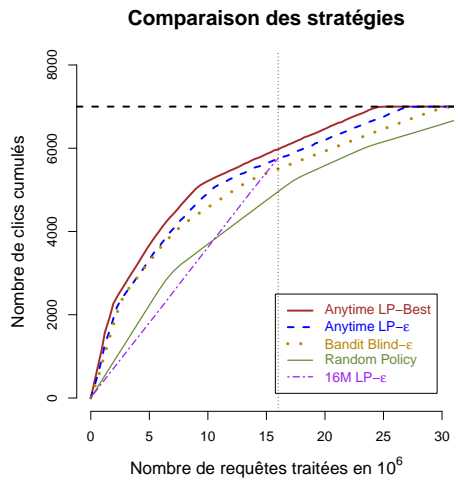


FIG. 1 – Simulations sans apparition ni disparition de Publicités. La courbe la plus élevée (continue et en gras « Anytime Lp-Best x ») est la stratégie optimale en supposant connues les probabilités de clic, ce qui est impossible. La courbe « Random Policy » correspond à une allocation au hasard.

tés (excepté par le fait que quand le budget est atteint, la publicité ne peut plus être choisie).

La figure 1 donne l'espérance du nombre de clics de chacune de ces techniques en fonction du nombre de requêtes. On peut être surpris par la relativement bonne performance de la stratégie aléatoire pure. Cette bonne performance s'explique par les probabilités qui sont dans une large mesure réparties un peu uniformément et par la présence d'un grand nombre de profils pour lesquels on ne dispose d'aucune information. Après environ 7 millions de visiteurs, et jusqu'à l'écoulement total des publicités, « Anytime LP- ϵ » se révèle plus performant que « Bandit Blind- ϵ ». Cela signifie que qu'au delà de 7 millions de requêtes, il est pertinent de tenir compte des limites de budget des publicités et donc de ne pas toujours servir la publicité avec l'espérance de clic la plus haute. L'amélioration est d'environ 5% et correspond dans ce

Afin d'illustrer certains aspects du problème, il est plus facile de commencer par supposer que les publicités n'ont pas de limite de temps et qu'il n'y a pas apparition de nouvelles publicités au cours du temps. Notre premier objectif est de quantifier le gain obtenu par l'introduction de la programmation linéaire et donc par la prise en compte des budgets maximaux des publicités.

Anytime LP-Best est une stratégie optimale calculée par PL dans le cas où les $p_{i,j}$ sont connus. Nous l'utilisons pour avoir une borne sur les performances possibles.

Random policy fournit une borne inférieure sur les performances. Elle choisit uniformément une publicité parmi celles dont le budget restant est non nul.

Anytime LP- ϵ est notre approche qui mélange PL et exploration. On utilise ici un ϵ -glouton. Dans ces expériences, ϵ a été choisi empiriquement à 0,92.

Bandit Blind- ϵ utilise plusieurs bandits agissant indépendamment les uns des autres. Cela signifie que chacun essaie de servir à chaque profil la publicité avec la probabilité de clic la plus élevée. On ne tient pas compte du budget de chacune des publicités

contexte à la moitié de l'écart à la courbe optimale (celle qui connaît toutes les probabilités). La faiblesse des stratégies basées sur la programmation linéaire provient du fait qu'elles ont à tout moment besoin de savoir combien il y aura de visites de chaque profil. Or, il semble difficile de prévoir ce nombre avec une grande précision. Dans la figure 1, les deux courbes PL ne sont pas des performances issues d'une seule simulation. En fait, chaque nombre de visiteurs fourni en abscisses, correspond à une PL effectuée en connaissant ce nombre de visiteurs. On peut s'intéresser par exemple à l'évolution du nombre de clics pour un PL prévoyant 16.10^6 de visiteurs au cours du temps. Cela est représenté par la courbe « 16M Simplex- ε ». On s'aperçoit que la performance est quasiment linéaire au cours du temps. Cela montre que notre algorithme identifie assez rapidement les allocations à réaliser sur les publicités importantes et tient bien compte des budgets. De plus, cela est intéressant car dans ce contexte, les publicités sont écoulees avec une vitesse quasi-constante au cours du temps. Par contre, il faut garder à l'esprit que si l'on a une estimation trop peu précise du nombre de visites, alors on peut faire beaucoup moins bien qu'une approche plus gloutonne comme les bandits parallèles.

5.3 Dans un environnement dynamique

Nous nous intéressons maintenant au rôle de l'horizon H discuté dans la sec. 4.2. Pour cela, nous introduisons un modèle génératif des publicités. À chaque pas de temps, une nouvelle publicité peut apparaître avec une probabilité notée u . De plus, les publicités ont maintenant une durée de vie fixée. La figure 2 illustre les performances obtenues par différents réglages de H pour deux valeurs de la durée de vie L . Elle représente l'espérance du nombre de clics par unité de temps. Le début de la courbe correspond à l'écoulement du stock initial de publicités avant de se placer dans une sorte de régime stationnaire. Dans ce régime, les stratégies utilisant la PL sont systématiquement au-dessus des méthodes de bandit (Pour $H = 1$, MAB+LP est en fait un Bandit). Le gain de performance est de l'ordre de 4%. Il existe une valeur optimale pour l'horizon (autour de $\frac{L}{2}$). Elle correspond à un équilibre entre l'apparition des nouvelles publicités et leur vitesse d'écoulement. Notons que plus on augmente la durée de vie des publicités, plus l'horizon optimal est grand. Dans le cas où la durée de vie est infinie, le plus grand horizon possible est le meilleur.

6 Conclusion et perspectives

Nous avons étudié le problème de l'optimisation du nombre de publicités cliquées sur un site *web*. Nous avons abordé le problème dans un cas plus proche de la réalité que celui étudié classiquement, en particulier du point de vue des contraintes de temps et de budget sur les publicités. Dans ce contexte, nous avons développé une approche hybride (MAB+LB) qui réalise une exploration tout en tenant compte des contraintes lors de sa phase d'exploitation. Les performances obtenues sont proches de celles des stratégies connaissant *a priori* les probabilités de clic et sont dans tous les cas supérieures aux bandits lancés en parallèle. De plus notre approche possède des aspects intéressants en pratique :

- nous avons tendance à écouler les publicités à une vitesse constante au cours de sa durée de vie. Ce point est généralement très apprécié des annonceurs.
- Il est possible d'estimer le gain obtenu à partir d'un nombre donné de profils. Il suffit pour cela de relancer une planification en éliminant les profils concernés. Cela est in-

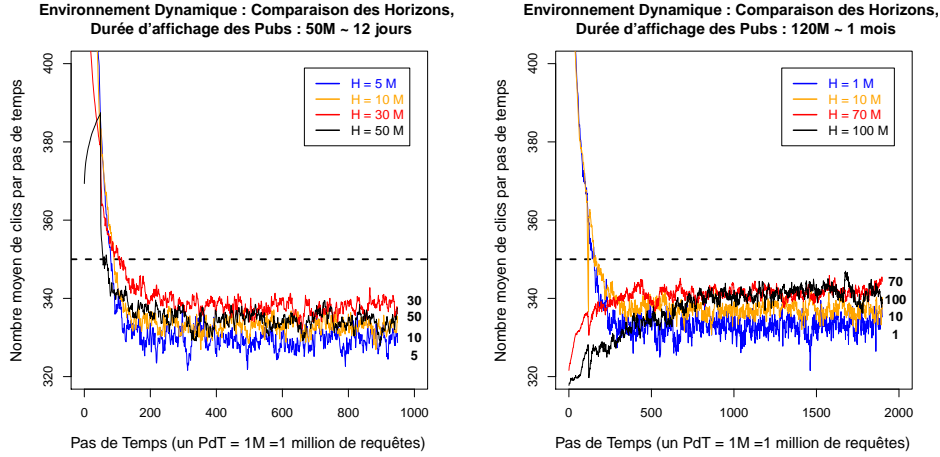


FIG. 2 – Illustration du rôle de l'horizon H dans un environnement dynamique avec durée de vie finie et apparition de nouvelles publicités, moyenne sur 1000 simulations. La fréquence d'apparition d'une nouvelle publicité a été choisie de façon à ce qu'il ne soit pas possible d'écouler toutes les publicités. Pour rester réaliste, on ne dispose pas non plus d'une surabondance de publicités. Le choix de l'horizon optimal est aussi lié à la durée de vie des publicités. Une nouvelle répartition optimisée par PL est faite toutes les 10^4 visites.

intéressant dans le cas où les revenus proviennent en partie d'une vente en coût par clic (CPC) et d'autre part d'une vente plus classique à l'impression (CPM).

Plusieurs extensions sont d'ores et déjà envisagées à ce travail. Tout d'abord, un déploiement pour la régie publicitaire d'Orange Labs va être réalisé ; cela permettra notamment de tester notre approche face à de véritables internautes. Notons qu'il subsiste également de fortes interrogations dans le cas où plusieurs publicités sont affichées simultanément sur une page web (l'hypothèse d'indépendance devenant clairement très fausse). Enfin, nous n'avons pas mentionné l'apprentissage par renforcement (RL) comme une approche possible pour résoudre ce problème. Hors, il est clair que nous faisons face ici à ce type de problèmes ; nous avons écarté les algorithmes traditionnels de RL du fait de la complexité de l'espace d'états (très grand, discret, mal structuré) et du nombre importants d'actions possibles. Notons qu'à l'inverse, l'algorithme que nous proposons ici (MAB+LP) peut très bien s'appliquer à tout un ensemble de problèmes de RL à horizon fini et variant dans le temps ; c'est plutôt cette voie que nous envisageons d'étudier par la suite.

Remerciements

Ce travail a été partiellement financé par les Orange Labs *via* le contrat de recherche externalisé numéro 46 146 063 - 8360. Le modèle et ses paramètres utilisés dans cet article ont été conçus avec et validés par Orange Labs afin de conserver les caractéristiques essentielles du problème réel. Nous remercions tout particulièrement Fabrice Clérot et Stéphane Sénécal pour les riches et nombreuses discussions que nous avons eues tout au long de ce travail.

Références

- Abe, N. et A. Nakamura (1999). Learning to Optimally Schedule Internet Banner Advertisements. In *Proc. 16th ICML*, pp. 12–21. Morgan Kaufmann Publishers Inc.
- Auer, P., N. Cesa-Bianchi, et P. Fischer (2002). Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning Journal* 47(2-3), 235–256.
- Chakrabarti, D., R. Kumar, F. Radlinski, et E. Upfal (2008). Mortal Multi-Armed Bandits. In D. K. et al. (Ed.), *Proc. NIPS*, pp. 273–280. MIT Press.
- Gabillon, V. (2009). Machine Learning Tools for On-line Advertisement. Technical report, INRIA Lille - Nord Europe, École Normale Supérieure de Cachan, TELECOM SudParis.
- Granmo, O.-C. (2008). A bayesian learning automaton for solving two-armed Bernoulli bandit problems. In *Proc. 7th ICML-A*, pp. 23–30. IEEE Computer Society.
- Kakade, S. M., S. Shalev-Shwartz, et A. Tewari (2008). Efficient Bandit Algorithms for Online Multiclass Prediction. In *Proc. 25th ICML*, pp. 440–447. ACM.
- Langford, J. et T. Zhang (2008). The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information. In J. P. et al. (Ed.), *Proc. NIPS*, pp. 817–824. MIT Press.
- Mahdian, M. et H. Nazerzadeh (2007). Allocating Online Advertisement Space with Unreliable Estimates. In *Proc. ACM Conference on Electronic Commerce*, pp. 288–294.
- Mehta, A., A. Saberi, U. Vazirani, et V. Vazirani (2005). Adwords and Generalized On-line Matching. In *Proc. 46th FOCS*, pp. 264–273. IEEE Computer Society.
- Pandey, S., D. Agarwal, D. Chakrabarti, et V. Josifovski (2007). Bandits for Taxonomies : A Model-based Approach. In *Proc. 7th SIAM-DM*.
- Pandey, S. et C. Olston (2006). Handling Advertisements of Unknown Quality in Search Advertising. In B. S. et al. (Ed.), *Proc. NIPS*, pp. 1065–1072. MIT Press.
- Robbins, H. (1952). Some Aspects of the Sequential Design of Experiments. *Bulletin of the American Mathematics Society* 58, 527–535.
- Wang, C.-C., S. Kulkarni, et H. Poor (2005). Bandit Problems With Side Observations. *IEEE Transactions on Automatic Control* 50(3), 338–355.

Summary

We consider the problem of displaying commercial ads on web pages. More and more advertisers are willing to pay only when their ads are clicked. In this emerging “cost per click” model, the ad server has to learn the interest of each type of visitors for the different ads in order to maximize the income. This “exploration versus exploitation” problem is often addressed as if resources were unlimited whereas in a realistic context, budget constraints on the ads like limited number of clicks, as well as the duration of the ad campaigns, should be taken into account. For this purpose, we introduce MAB+LP, a hybrid approach based on linear programming and multi-armed bandits to handle this problem and investigate its performance through simulations on a realistic model designed with an important commercial web actor. These experiments exhibit near optimal performance on the investigated problem setting.