Managing advertising campaigns – an approximate planning approach

Sertan Girgin, Jérémie Mary, Philippe Preux, Olivier Nicol

Front. Comput. Sci.
DOI 10.1007/s11704-012-2873-5

RESEARCH ARTICLE

Managing advertising campaigns – an approximate planning approach

Sertan GIRGIN[1,2], **Jérémie MARY**[1,2], **Philippe PREUX**[1,2], **Olivier NICOL**[1,2]

1 Team-Project Sequel, INRIA Lille Nord Europe, Villeneuve d'Ascq 59650, France
2 LIFL (UMR CNRS), Université de Lille, Villeneuve d'Ascq 59650, France

Abstract We consider the problem of displaying commercial advertisements on web pages, in the "cost per click" model. The advertisement server has to learn the appeal of each type of visitors for the different advertisements in order to maximize the profit. Advertisements have constraints such as a certain number of clicks to draw, as well as a lifetime. This problem is thus inherently dynamic, and intimately combines combinatorial and statistical issues. To set the stage, it is also noteworthy that we deal with very rare events of interest, since the base probability of one click is in the order of 10^{-4}. Different approaches may be thought of, ranging from computationally demanding ones (use of Markov decision processes, or stochastic programming) to very fast ones. We introduce NOSEED, an adaptive policy learning algorithm based on a combination of linear programming and multi-arm bandits. We also propose a way to evaluate the extent to which we have to handle the constraints (which is directly related to the computation cost). We investigate performance of our system through simulations on a realistic model designed with an important commercial web actor.

Keywords advertisement selection, web sites, optimization, non-stationary setting, linear programming, multi-arm bandit, click-through rate (CTR) estimation, exploration-exploitation trade-off.

Received June 15, 2011; accepted November 6, 2011

E-mail: sertan.girgin@inria.fr

1 Introduction

The ability to efficiently select items that are likely to be clicked by a human visitor of a web site is a very important issue. Whether for the mere comfort of the user to be able to access the content he/she is looking for, or to maximize the income of the website owner, this problem is strategic. The selection is based on generic properties (date, world news events, ...), along with available personal information (ranging from mere IP related information to more dedicated information based on the login to an account). The scope of applications of this problem ranges from advertisement or news display (see for instance the Yahoo! Front Page Today Module), to web search engine result display. There are noticeable differences between these examples: in the first two cases, the set of items from which to choose is rather small, in the order of a few dozens; in the latter case, the set contains billions of items. The lifetime of items may vary considerably, from a few hours for news, to weeks for web advertisements, to years for pages returned by search engine. Finally, the objective ranges from drawing attention and clicks on news, to providing the most useful information for search engines, to earning a maximum of money in the case of advertisement display. Hence, it seems difficult to consider all these settings at once and in this paper, we consider the problem of selecting advertisements, in order to maximize the profit earned from clicks: we consider the "cost to click" economic model in which each single click on an advertisement brings a certain profit. We wish to study principled approaches to solve this problem in the most realistic setting; for that purpose, we

consider the problem with:

- finite amounts of advertising campaigns,
- finite amounts of clicks to gather on each campaign,
- finite campaign lifetimes,
- the appearance and disappearance of campaigns along days, and
- a finite flow of visitors and page requests.

With these assumptions, we would like to emphasize that our goal is not to optimize any asymptotic behavior and exhibit algorithms that are able to achieve optimal asymptotic behavior (but perform badly for much too long). To the opposite, we concentrate on the practical problem faced here and now by the web server owner: he/she wants to make money now, and do not really care about ultimately becoming a billionaire when the universe will have collapsed (which is likely to happen in a not so remote future with regards to asymptotic times either). In the same order of ideas, we also want to keep the solution computable in "real"-time, real meaning here within a fraction of a second, and able to support the high rate of requests observed on the web server of an important web portal. Of course, such requirements impede the quality of the solution, but these requirements are necessary from the practical point of view; furthermore, since we have to deal with a lot of uncertainty originating from various sources, the very notion of optimality is quite relative here.

In Section 2, we formalize the problem under study, and introduce the vocabulary and the notation used throughout the paper. Our notations are summarized in an appendix to the paper. The problem we tackle is actually changing over time; for pedagogical reasons, in Section 3 we first study the problem under a static setting where the set of advertising campaigns are known in advance and the time horizon is fixed, before moving to the more general dynamic setting without these constraints in Section 4. We define a series of problems of increasing complexity, ranging from the case in which all information is available, to the case where key information is missing. Assessing algorithms in the latter one is difficult, in particular from a methodological point of view, and spanning this range of problems let us assess our ideas in settings in which there is a computable optimal solution against which the performance of algorithms may be judged. Section 5 presents related works. Section 6 presents some experimental results of our algorithm near optimal sequential estimation and exploration for decision (NOSEED) in both static and dynamic settings. Finally, Section 7 concludes and we briefly discuss the lines of foreseen future works.

2 Formalization of the problem

At a given time t, there is a pool of advertising campaigns. Each advertising campaign in the pool has a starting time, a lifetime and a click budget that is expected to be fulfilled during its lifetime. At each click on an advertisement of the campaign, a certain profit is made. The status of an advertising campaign can be either one of the following (Fig. 1):

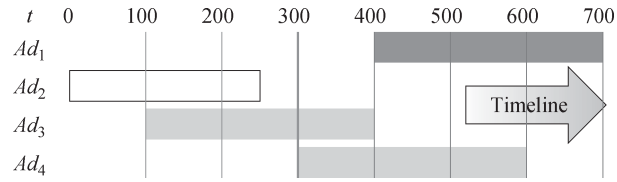

Fig. 1 At time $t = 300$, Ad_1 is in scheduled state (in dark grey), Ad_2 has expired (in white), Ad_3 and Ad_4 are running with remaining lifetimes of 100 and 300, respectively (in light grey)

scheduled when the campaign will begin at some time in the future,

running when the campaign has started but not expired yet,

expired when either the lifetime of a campaign has ended or the click budget has been reached.

The advertisements of a campaign can only be displayed when it is in the *running* state.

Each advertising campaign is assumed to have a unique identifier, and we will represent an advertising campaign by a tuple (S, L, B, cp, rb) where S, L, B and cp denote its starting time, lifetime, click budget and profit per click, respectively; $rb \leqslant B$ denotes the remaining click budget of the advertising campaign. Note that, for a given advertising campaign, all parameters except the remaining click budget are constant; the remaining click budget is initially equal to the click budget of the advertising campaign and decreases with time as it receives clicks from the visitors. Throughout the paper, the advertising campaign with identifier k will be denoted by Ad_k and its parameters will be identified by subscript k, e.g., S_k will denote the starting time of the advertising campaign with identifier k.

Now, the problem that we are interested in is as follows (Fig. 2):

- The web site receives a continuous stream of page requests. Each request originates from a "visitor", that is,

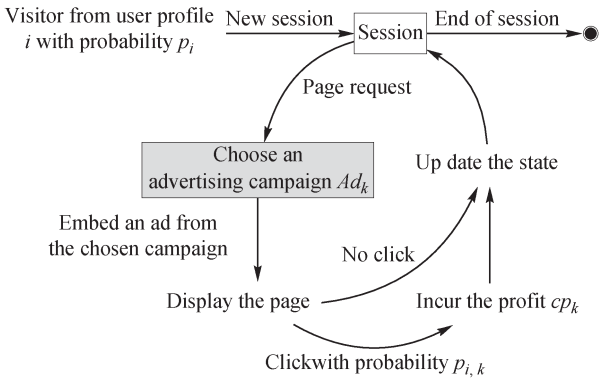

Fig. 2 The interaction between a visitor and the system

a human being browsing in some way the website. We assume that non human visitors (robots) may be identified as such, and are filtered out[1]. Each visitor is assumed to belong to one among N possible user profiles; the user profiles are numbered from 1 to N. We will use U_i to denote the i^{th} user profile and v_i to denote the probability that a visitor belongs to that user profile[2].

- When a visitor visits the web site, a new "session" begins and we observe one or several iterations of the following sequence of events:
  - The visitor requests a certain page of the web site.
  - The requested page is displayed to this visitor with an advertisement from advertising campaign with identifier k embedded in it.
  - The visitor clicks on the advertisement with a certain probability $p_{i,k}$ where i denotes the user profile of the visitor; this probability is usually called the click-through rate (CTR) and the event itself is a Bernoulli trial with success probability $p_{i,k}$.
  - If there is a click, then the profit associated with the advertising campaign is incurred.
- After a certain number of page requests, the visitor leaves the web site and the session terminates.

Returning visitors do not change the nature of the problem given that the session information persists, and for the sake of simplicity we will be assuming that there are no returning visitors.

The objective is to maximize the total profit by choosing the advertisements to be displayed "carefully". Since page requests are atomic actions, in the rest of the paper we will take a page request as the *unit of time* to simplify the discussion, i.e., a time step will denote a page request and vice versa. Note that in the real-world, some of the parameters mentioned above may not be known with certainty in advance. For example, we do not know the visit probabilities of the user profiles, their probability of click for each advertising campaign, the actual profiles of the visitors, or the number of requests that they will make; the number of visitors may change with time and new advertising campaigns may begin. These and other issues that we will address in the following sections of the paper make this problem a non-trivial one to solve.

3 Static setting

In order to better understand the problem and derive our solution, we will first investigate it under a static setting. In this setting, we assume that

- there is a pool of K advertising campaigns, and
- the properties of the advertising campaigns (i.e., their starting times, lifetimes, click budgets and click profits) are known in advance.

Note that, this leads to a fixed time horizon T which is equal to the latest ending time of the advertising campaigns. At time T, the task is finished. Other parameters of the problem, such as the click and visit probabilities, may or may not be known with certainty. We will start with the case in which all the information is available, and subsequently move to the setting in which uncertainty comes into play, and then only a part of the information will be available.

3.1 Static setting with full information

In the static setting with full information, we assume that all parameters are known. To be more precise:

(a) the visit probabilities of user profiles, v_i, and their click probabilities for each advertising campaign, $p_{i,k}$ are known, and,

(b) there is no uncertainty in the actual profiles of the visitors, i.e., we know for sure the profile of each visitor.

Note that, even if we have full information, the visitor at time t and whether the visitor will click on the displayed advertisement or not are still unknown.

We first define the problem we wish to solve as a Markov

[1] The identification of robots is not necessarily easy to perform, but even if some of them are not filtered out, that would not bring serious problems to our study.
[2] The sum of v_i over all user profiles is equal to 1, which forms a categorical distribution with probability mass function $f_{\mathbf{P}}(U_i) = v_i$
[3] To be precise, one optimal solution, or a set of equally performing optimal solutions.

decision process (MDP) [1]. This implies that the problem under consideration has an optimal solution[3]. For the sake of clarity, we only detail the MDP formulation in the static setting, but subsequent, more complex settings, may easily be cast into the MDP (or partially-observable MDP) framework. Then, we consider the problem of determining an optimal policy for this MDP. As it will be shown in the following section, the state space of the MDP grows linearly with time and exponentially with the number of advertising campaigns. This huge state space makes it difficult to determine an optimal policy by straightforward dynamic programming approaches in a reasonable time for any practical application; this raises the question of whether there can be other approaches to solve the problem and obtain a policy that performs "well" (also this explains why we do not solve the problem using traditional MDP algorithms).

Being interested in real settings, thus looking for non asymptotic performance, and wishing to have an algorithm that performs as best as possible in an efficient way, we examine various issues and subsequently propose the NOSEED algorithm which aims to handle them both in simple and more complex settings as will be detailed in Sections 3.2, 3.3, and 4.

3.1.1 The underlying Markov decision problem for the advertising selection problem

At any time t, the state of this version of the problem can be fully represented by a tuple that consists of time t, the time horizon T, the visit and click probabilities, and a set of tuples denoting the advertising campaigns:

$$\langle t, T, \{v_i\}, \{p_{i,k}\}, Ad_1 = \langle S_1, L_1, B_1, cp_1, rb_1\rangle, \ldots, Ad_K\rangle.$$

By omitting the fixed parameters, this tuple can be more compactly represented as $\langle t, rb_1, \ldots, rb_K\rangle$.

Given a state $s = \langle t, rb_1, \ldots, rb_K\rangle$, if there is no click at that time step or there is no running advertising campaign then the next state, which we will denote by $s'_{noclick}$, has the same representation as s except the t component since click budgets of campaigns do not change, i.e., $s'_{noclick} = \langle t+1, rb_1, \ldots, rb_K\rangle$. In case an advertisement from a running advertising campaign Ad_k is clicked, the remaining click budget of Ad_k will be reduced by 1 and the next state becomes $\langle t+1, rb_1, \ldots, rb_k - 1, \ldots, rb_K\rangle$; we will denote this state by $s'_{click,k}$.

A policy is defined as a mapping from states to a distribution over the set of advertising campaigns; given a particular state, the policy determines which advertising campaign to display at that state. A policy is called optimal if it maximizes the expected total profit. Let $V(s)$ denote the expected total profit that can be obtained by following an optimal policy starting from state s until the end of time horizon; $V(s)$ is usually called the value of state s. Now, suppose that there is a visitor from the i^{th} user profile at state s; the expected total profit that can be obtained by displaying an advertisement from a running advertising campaign Ad_k can be defined as:

$$V_{i,k}(s) = p_{i,k}[cp_k + V(s'_{click,k})] + (1 - p_{i,k})V(s'_{noclick}), \quad (1)$$

and the optimal policy, i.e., the best advertising campaign to display, would be to choose advertising campaign with the maximum expected total profit, i.e., $\arg\max_{Ad_k} V_{i,k}(s)$. Note that, the value of state s can be calculated by taking the expectation of maximum $V_{i,k}(s)$ values over all user profiles and we have:

$$V(s) = \sum_{U_i} \max_{Ad_k} V_{i,k}(s). \quad (2)$$

Regarding expired campaigns, we define their value to be 0. Using Eqs. (1) and (2), the value of any state can be determined, for example, by dynamic programming; henceforth, the optimal policy can be determined too. However, the size of the state space is equal to $(T - t) \times rb_1 \ldots \times rb_K$ and grows exponentially with the number of advertising campaigns (with order equal to their budgets). From a practical point of view, this huge state space makes such solutions very computationally demanding, and unable to meet our requirements in this regard.

3.1.2 A greedy approach

When we look at Eq. (1) more carefully, it is easy to see that the value of the next state without a click, $V(s'_{noclick})$, is an upper bound for the value of the next state with a click, $V(s'_{click,k})$. Replacing the second term by $V(s'_{noclick]}) - \xi_{i,k}$ where $\xi_{i,k}$ is a constant that depends on s, Ad_k and the user profile U_i, we obtain:

$$V_{i,k}(s) = p_{i,k}cp_k - p_{i,k}\xi_{i,k} + V(s'_{noclick}). \quad (3)$$

If $\xi_{i,k}$ values are small compared to the corresponding click profits, i.e., their effect is negligible, or they are ignored, then the optimal policy becomes choosing the advertising campaign with the highest expected profit per click among the set of running campaigns at that state denoted by C:

$$\begin{aligned}\arg\max_{Ad_k \in C} V_{i,k}(s) &= \arg\max_{Ad_k \in C} [p_{i,k}cp_k + V(s'_{noclick})] \\ &= \arg\max_{Ad_k \in C} p_{i,k}cp_k.\end{aligned}$$

We will call this particularly simple method the highest expected value (HEV) policy. Alternatively, we can employ a stochastic selection method where the selection probability of a running advertising campaign is proportional to its expected profit per click. This variant will be called the stochastic expected value (SEV) policy.

As both policies exploit advertising campaigns with possibly high return and assign lower priority to those with lower return, one expects them to perform well if the lifetimes of the advertising campaigns are "long enough" to ensure their click budgets. However, they may show inferior performances even in some trivial situations, that is $\xi_{i,k}$ terms are significant. For example, assume that there is a single user profile and two advertising campaigns, Ad_1 and Ad_2, starting at time $t = 0$ with click probabilities of 0.005 and 0.01, lifetimes of $L_1 = 2\,000$ and $L_2 = 4\,000$ time steps, budgets of $B_1 = 10$ and $B_2 = 20$ clicks, and unit profits per click i.e., $cp_1 = cp_2 = 1$ (Fig. 3). In this particular case, starting from $t = 0$, HEV policy always chooses Ad_2 until this campaign expires (on expectation at $t = 2\,000$, at which point the other campaign Ad_1 also expires) and this results in an expected total profit of 20 units; SEV policy displays on average twice as many advertisements from Ad_2 compared to Ad_1 during the first 2000 time steps, and performs slightly better with an expected total profit of $23\frac{1}{3}$. However, both figures are less than the value of 25 that can be achieved by choosing one of the campaigns randomly with equal probability. Note that, by displaying advertisements from only Ad_1 in the first 2000 time steps until it expires and then Ad_2 thereafter, it is possible to obtain an expected total profit of 30 that satisfies the budget demands of both advertising campaigns; the lifetime of Ad_2, which is long enough to receive a sufficient number of clicks with the associated click probability, allows this to happen. In order to derive this solution, instead of being short-sighted, it is compulsory to take into consideration the interactions between the advertising campaigns over the entire timeline and determine which advertising campaign to display accordingly, in other words, consider a planning problem, as in the dynamic

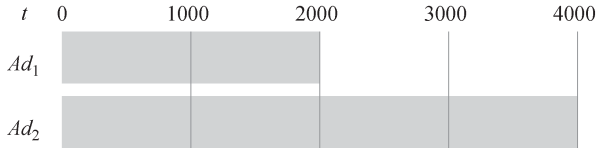

Fig. 3 A toy example in which HEV and SEV policies have suboptimal performance. Ad_1 and Ad_2 have the same unit profit per click, click probabilities of 0.005 and 0.01, and total budgets of $B_1 = 10$ and $B_2 = 20$ clicks, respectively. The expected total profits of HEV and SEV are 20 and $23\frac{1}{3}$ compared to a maximum achievable expected total profit of 30

programming solution mentioned before.

Observing Fig. 3, it is easy to see that the interactions between the advertising campaigns materialize as *overlapping*

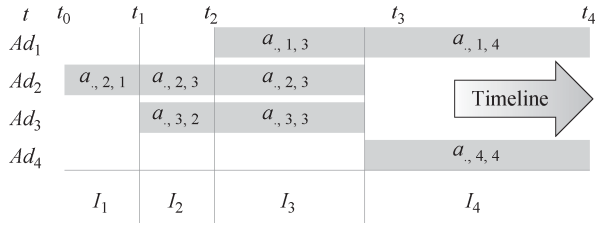

Fig. 4 The timeline divided into intervals and parts. I_j denotes the j^{th} interval $[t_{j-1}, t_j]$ and $a_{k,j}$ denotes the allocation for advertising campaign Ad_k in interval I_j. The first index of a (user profile) is left unmentioned for the sake of clarity. In this particular example, the set of running advertising campaigns in the second interval is $AI_2 = \{Ad_2, Ad_3\}$, and the set of intervals that cover Ad_1 is $IA_1 = \{I_3, I_4\}$

time intervals over the timeline[4]; in this toy example the intervals are $I_1 = [0, 2\,000]$ and $I_2 = [2\,000, 4\,000]$, and what we are trying to find is the *optimal allocation* of the number of advertising campaign displays in each interval. This can be posed as the following optimization problem where $a_{k,j}$ denotes the number of displays allocated to Ad_k in the interval I_j:

$$\begin{aligned} \text{maximize} \quad & 0.005 \times a_{1,1} + 0.01 \times (a_{2,1} + a_{2,2}), \\ \text{s.t.} \quad & a_{1,1} + a_{2,1} \leqslant 2\,000, a_{2,2} \leqslant 2\,000, \\ & 0.005 \times a_{1,1} \leqslant 10, 0.01 \times (a_{2,1} + a_{2,2}) \leqslant 20, \end{aligned}$$

which has an optimal solution of $a_{1,1} = a_{2,2} = 2000$ and $a_{2,1} = 0$. One can then use this optimal allocation to calculate the display probabilities for both advertising campaigns proportional to the number of displays allocated to them in the corresponding time intervals.

3.1.3 Optimal allocation approach

Let E_k be the ending time of advertising campaign Ad_k, which is simply equal to the sum of its starting time and lifetime. Given a pool of K advertising campaigns C, the time intervals during which the advertising campaigns overlap with each other can be found from the set of their starting and ending times. Let $t_0, t_1, \ldots, t_M$, $M \leqslant 2 \times K$, be the sorted list of elements of the set of starting and ending times of the advertising campaigns; without loss of generality, we will assume that $t_0 = 0$ as otherwise there will not be any advertising campaigns to display until t_0. By definition, the M intervals defined by $I_j = [t_{j-1}, t_j], 1 \leqslant j \leqslant M$ cover the entire timeline of the pool of the advertising campaigns. Let $AI_j = \{Ad_k | S_k < t_j \leqslant E_k\}$ be the set of running advertising

[4] See Fig. 4 for a more detailed example.

campaigns in interval I_j. Note that for some of the intervals, this set may be empty; these intervals are not of our interest as there will be no advertising campaigns to display during such intervals (which is certainly not good for the web site) and we can ignore them. Let $\mathcal{A} = \{I_j | AI_j \neq \emptyset\}$ be the set of remaining intervals, $l_j = t_j - t_{j-1}$ denote the length of interval I_j, and $IA_k = \{I_j | Ad_k \in AI_j\}$ be the set of intervals that cover Ad_k (Figure 4). Generalizing the formulation given above for the simple example and denoting the number of displays allocated to Ad_k in the interval I_j for the user profile U_i by $a_{i,k,j}$, we can define the optimization problem that we want to solve as follows:

$$\text{maximize} \quad \sum_{I_j \in \mathcal{A}} \sum_{Ad_k \in AI_j} cp_k p_{i,k} a_{i,k,j}, \tag{4}$$

$$\text{s.t.} \quad \sum_{Ad_k \in AI_j} a_{i,k,j} \leqslant v_i l_j, \quad \forall\, U_i, I_j \in \mathcal{A}, \tag{5}$$

$$\sum_{U_i} \sum_{I_j \in IA_k} p_{i,k} a_{i,k,j} \leqslant rb_k, \quad \forall\, Ad_k \in C. \tag{6}$$

The objective function (Eq. 4) aims to maximize the total expected profit, the first set of constraints (Eq. (5)) ensures that for each interval we do not make an allocation for a particular user profile that is over the capacity of the interval (i.e., the portion of the interval proportional to the visit probability of the user profile), and the second set of constraints (Eq. (6)) ensures that we do not exceed the remaining click budgets. This corresponds to the maximization of a *linear objective function* ($a_{i,k,j}$ being the variables), subject to *linear inequality constraints*, which is a *linear programming* problem. This problem can be solved efficiently using the simplex algorithm, or an interior-point method, or an other existing large scale approach if necessary. The number of constraints in the linear program is of order $O(NK)$ where N is the number of user profiles and K is the number of advertising campaigns, and the number of variables is of order $O(NK^2)$.

The solution of the linear program, i.e., the assignment of values to $a_{i,k,j}$, indicates the number of displays that should be allocated to each advertising campaign for each user profile and in each interval, but it does not provide a specific way to choose the advertising campaign to display to a particular visitor from user profile U_i at time t. For this, we need a method to calculate the display probability of each running advertising campaign from their corresponding allocated number of displays.

Let $\hat{a}_{i,k,j} = a_{i,k,j} / \sum_{Ad_k \in AI_j} a_{i,k,j}$ be the ratio of the allocation for user profile U_i and advertising campaign Ad_k in interval I_j to the total number of allocations for that user profile in the same interval. One can either pick the advertising campaign having the highest ratio in the first interval, i.e., $\arg\max_k \hat{a}_{i,k,0}$, which we will call the highest LP policy (HLP), or employ a stochastic selection method similar to SEV in which the selection probability of a campaign Ad_k is proportional to its ratio $\hat{a}_{i,k,0}$, which will be called the stochastic LP policy (SLP); SLP introduces certain degree of exploration which will be useful in more complex settings. Note that, as we are planning for the entire timeline, the solution of the linear program at time t may not allocate any advertising campaigns to a particular user profile i, i.e., it may be the case that $a_{i,k,j} = 0$ for all k, simply suggesting not to display any advertisement to a visitor from that user profile. In practice, when the current user is from such a user profile, choosing an advertising campaign with a low (or high) expected profit per click would be a better option and likely to increase the total profit at the end.

3.1.4 NOSEED: a two-phases, alternating algorithm

By defining and solving the linear program at each time step $0 \leqslant t < T$ for the current pool of non-expired advertising campaigns (which depends on the visitors that have visited the web site up until that time step, the advertising campaigns displayed to them and visitors' reactions to those displays), and employing one of the policies mentioned above, advertising campaigns can be displayed in such a way that the total expected profit is maximized, ignoring the uncertainty in the predictions of the future events (we will subsequently discuss the issues related to uncertainty).

When the number of advertising campaigns, and consequently the number of variables and constraints, is large, or when there is a need for fast response time, solving the optimization problem at each time step may not be feasible. An alternative approach would be to solve it regularly, for example, at the beginning for each interval or when an advertising campaign fulfills its click budget, and use the resulting allocation to determine the advertising campaigns to be displayed until the next resolution. In short, the algorithm alternates planning, with exploitation of this planning during multiple steps. This can be accomplished by updating the allocated number of advertising campaign displays as we move along the timeline, reducing the allocation of the chosen advertising campaigns in the corresponding intervals, and calculating the ratios that determine the advertising campaign to be displayed accordingly[5)]. Note that in practice, the planning step and the exploitation step can be asynchronous as long

[5)] The complete algorithm can be found in the appendix.

as the events that have occurred from the time that planning has started until its end are reflected properly to the resulting allocation. Such an algorithm belongs to the approximate dynamic programming family.

3.2 Dealing with uncertainty in the static setting with full information

The static setting with full information has two sources of uncertainty:

(a) the user profiles of visitors are drawn from a categorical distribution, and

(b) each advertising campaign display is a Bernoulli trial with a certain probability, which is known, and the result is either a success (i.e., click) or a failure (i.e., no click).

The aforementioned linear program solution of the optimization problem focuses on what happens in the expectation. Following the resulting policy in different instances of the same problem[6] may lead to different realizations of the total profit that vary from its expected value (due to the fact that the number of visitors from each user profile and the number of clicks on the displayed advertising campaigns will not exactly match their expected values).

As a simple example, consider the case in which there is a single user profile and two advertising campaigns Ad_1 and Ad_2 both having the same unit profit per click and a lifetime of 10^5 time steps, click probabilities of 0.001 and 0.002, and total budgets of 50 and 100, respectively. The solution of the linear program would allocate 50 000 displays to each advertising campaign with an expected total profit of 150, thus satisfying the budget demands. Figure 5 shows the cumulative distribution of the total profit over 1 000 independent runs for this problem using the stochastic LP policy and solving the optimization problem once at the beginning. Although values that are equal to or near the expected total profit are attained in more than half of the runs, one can observe a substantial amount of variability. In reality, reducing this variability may also be important and could be considered as a secondary objective to obtaining a high total profit. For the given example, slightly increasing the display probability of Ad_2 and decreasing that of Ad_1 would enable the accomplishment of this objective by preventing the risk of receiving fewer clicks than expected for Ad_2 without considerably compromising the outcome as the same risk also exists for Ad_1. This leads to the question of how to incorporate risk-awareness to our formulation of the optimization problem.

When we look closely at the objective function and the constraints of the linear program (Eqs. (4)–(6)), we can identify two sets of expressions of the form $v_i l_j$ and $p_{i,k} a_{i,k,j}$; the first one denotes the expected number of visitors from user profile U_i during the time-span of interval I_j, and the second one denotes the expected number of clicks that would be received if the advertising campaign Ad_k is displayed $a_{i,k,j}$ times to the visitors from user profile U_i. Note that visits from a particular user profile U_i occur with a known average rate v_i, and each visit occurs independently of the time since the previous visit. Therefore, the number of such visits in a fixed period of time t can be considered a random variable having a Poisson distribution with parameter $\lambda = v_i t$ which is equal to the expected number of visits that occur during that time period. Similarly, the number of clicks that would be received in a fixed period of time if advertising campaign Ad_k is displayed to the visitors from user profile U_i can also be considered a random variable having a Poisson distribution with parameter $\lambda = p_{i,k} t$. Let $Po(\lambda)$ denote a Poisson-distributed random variable with parameter λ. Replacing $v_i l_j$ and $p_{i,k} a_{i,k,j}$ terms with the corresponding random variables, we can convert the linear program into the following stochastic optimization problem:

$$\max \quad \sum_{I_j \in \mathcal{A}} \sum_{Ad_k \in AI_j} cp_k \mathbb{E}[Po(p_{i,k} a_{i,k,j})], \tag{7}$$

$$\text{s.t.} \quad \sum_{Ad_k \in AI_j} a_{i,k,j} \leqslant Po(v_i l_j), \quad \forall\, U_i, I_j \in \mathcal{A}, \tag{8}$$

$$\sum_{U_i} \sum_{I_j \in IA_k} Po(p_{i,k} a_{i,k,j}) \leqslant rb_k, \quad \forall\, Ad_k \in C. \tag{9}$$

The summation of independent Poisson-distributed random variables also follows a Poisson distribution whose parameter is the sum of the parameters of the random variables. Assuming that $Po(p_{i,k} a_{i,k,j})$ are independent, the budget constraints in Eq. (9) can be written as:

$$Po\left(\sum_{U_i} \sum_{I_j \in IA_k} p_{i,k} a_{i,k,j}\right) \leqslant rb_k, \quad \forall\, Ad_k \in C, \tag{10}$$

which is equivalent to its linear program counterpart in expectation. The rationale behind this set of constraints is to bound the total expected number of clicks for each advertising campaign (while at the same time trying to stay as close as possible to the bounds due to maximization in the objective function). Without loss of generality, assume that in the optimal allocation the budget constraint of advertising campaign Ad_k is met. This means that the expected total number of clicks for Ad_k will be a Poisson-distributed random variable with parameter rb_k and in any particular instance of the problem the probability of realizing this expectation (our target) would be 0.5. In order to increase the likelihood of reaching

[6] An "instance" refers here to a certain realization of the random problem.

the target expected total number of clicks, a possible option would be to use a higher budget limit in the constraint. Let Λ_k be our risk factor[7] and $Po(\lambda_k)$ be the Poisson-distributed random variable having the smallest parameter λ_k such that $Pr(Po(\lambda_k) > rb_k - 1) \geqslant \Lambda_k$ which is equivalent to

$$1 - \Lambda_k \geqslant \mathbf{F}_{Po(\lambda_k)}(rb_k - 1),$$

where $F_{Po(\lambda_k)}$ is the cumulative distribution function of $Po(\lambda_k)$. Note that rb_k and Λ_k are known, and λ_k can be found using numerical methods. If we replace rb_k with λ_k in the budget constraint and solve the linear optimization problem again, the expected total number of clicks for Ad_k based on the new allocation would be greater than or equal to rb_k and will have an upper bound of λ_k. Following the same strategy, one can derive new bounds for the user profile constraints and replace $v_i l_j$ terms in Eq. (8) with the smallest value of $\lambda_{i,j}$ such that the Poisson-distributed random variable $Po(\lambda_{i,j})$ satisfies $1 - \Lambda_{i,j} \geqslant \mathbf{F}_{Po(\lambda_{i,j})}(v_i l_j)$ and $\Lambda_{i,j}$ is the risk factor. In this case, an additional set of constraints defined below is necessary to ensure that for each interval the sum of advertising campaign allocations for all user profiles do not exceed the length of the interval:

$$\sum_{U_i} \sum_{Ad_k \in AI_j} a_{i,k,j} \leqslant l_j, \quad \forall I_j \in \mathcal{A}\,. \tag{11}$$

As presented in Fig. 5, in our simple example using a common risk factor of 0.95 results in a cumulative distribution of total profit which is more concentrated toward the optimal value compared to the regular linear program approach.

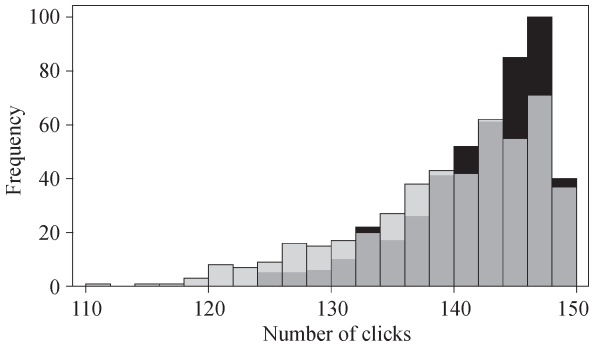

Fig. 5 The distribution of the total profit less than its expected value over 1000 independent runs on the toy example with two advertising campaigns; the dark shaded bars depict SLP with a risk factor of 0.95. In reality, the realization will be only one of the runs and therefore more concentration near the maximum value is better (see text for more explanation)

3.3 Static setting with partial information

In the settings discussed so far, we have assumed that two important sets of parameters, the visit probabilities of user profiles $\{U_i\}$ and their click probabilities for each advertising campaign $\{p_{i,k}\}$ are known. However, this is a rather strong assumption and in reality these probabilities are hardly known in advance; instead, they have to be estimated based on observations, such as the profiles of the existing visitors, the advertising campaigns that have been displayed to them and their responsive actions (i.e., whether they have clicked on a displayed advertisement or not). An accurate prediction of these probabilities results in the display of more attractive advertisements to the web site visitors.

Once this estimation problem is solved, one has to deal with probabilities to decide on which advertisement to display. This problem of decision making in face of uncertainty raises the exploration/exploitation dilemma, one having to balance the exploitation of what is already known, with the exploration of new, potentially better, decisions.

We discuss these two issues in the next two sections.

3.3.1 Estimating the probabilities

The simplest way to estimate unknown probabilities would be to use maximum likelihood estimation. In our problem, the profile of a visitor can be considered a categorical random variable $\mathbf{U}$ with profile U_i having an estimated visit probability of $\hat{v}_i$, and the click of a visitor from user profile U_i on an advertisement from advertising campaign Ad_k can be considered a Bernoulli random variable $\mathbf{p}_{i,k}$ with success probability $\hat{p}_{i,k}$.

Let $visit_i$ denote the total number of visitors from user profile U_i that have visited the web site at time $0 \leqslant t$, then the maximum likelihood estimate of $\hat{v}_i$ will be $visit_i/(t+1)$, and similarly the maximum likelihood estimate of $\hat{p}_{i,k}$ at time t will be $click_{i,k}/display_{i,k}$ where $click_{i,k}$ is the number of times that visitors from user profile U_i clicked on advertisement Ad_k and $display_{i,k}$ is the number of times Ad_k had been displayed to them[8]. Since $visit_i$ values are initially 0, the estimates will also be 0 until we observe a visit from the corresponding user profiles. In order remedy this situation, it is customary to assign a prior ϑ_i, e.g., 1, for each user profile and define $\hat{v}_i$ as

$$\hat{v}_i = \frac{visit_i + \vartheta_i}{t + 1 + \sum_{i=1}^{N} \vartheta_i}.$$

The priors of click probabilities can also be assigned in a similar manner. In practice, as the number of visits is high and the number of user profiles is low, the maximum likelihood estimates of visit probabilities will be quite accurate.

[7] Typical values include 0.90, 0.95, and 0.99.

[8] For brevity, the time indices have been dropped from $visit_i$, $display_{i,k}$ and $click_{i,k}$.

Alternatively, we can employ Bayesian maximum a posteriori estimates using the conjugate priors. The conjugate priors of the categorical and Bernoulli distributions are Beta and Dirichlet distributions, respectively. If $Beta(\alpha_{i,k},\beta_{i,k})$ is the Beta prior with hyper-parameters $\alpha_{i,k}$ and $\beta_{i,k}$ for click probability $p_{i,k}$, then the posterior at time t is the Beta distribution with hyper-parameters $\alpha_{i,j} + click_{i,k}$ and $\beta_{i,j} + display_{i,k} - click_{i,k}$. Setting both hyper-parameters to 1 corresponds to having a uniform prior. At time t, the posterior of the prior Dirichlet distribution with hyper-parameters v_i for $\mathbf{U}$ will have hyper-parameters $v_i + visit_i$. The initial hyper-parameters can be guessed or determined empirically based on historical data. As we will see later in the experiment section, choosing good priors may have a significant effect on the outcome.

By estimating probabilities at each time step (or periodically) and replacing the actual values with the corresponding estimates, we can use the approach presented in the previous section to determine allocations (optimal up to the accuracy of the estimations) and choose advertising campaigns to display. For maximum a posteriori estimates, the mode of the posterior distribution can be used as a point estimate and a single instance of the problem can be solved, or several instances of the problem can be generated by sampling probabilities from the posterior distributions, solved separately and then the resulting allocations can be merged (for example taking their mean; note that, in this case the final allocations will likely be not bound to the initial constraints).

3.3.2 Exploration-exploitation trade-off

As in many online learning problems, one important issue is the need for balancing the exploitation of the current estimates and exploration, i.e., estimation of the unknown or less-known (e.g., with higher variance) parameters. Using the solution of the optimization problem without introducing any additional exploration may introduce substantial bias to the results. This exploration/exploitation trade-off problem can be formulated as a multi-arm bandit problem (with the advertising campaigns in the role of arms). Based on the multi-arm bandit framework, exploration can be introduced to the allocation policy in various ways, among which we mention the following two:

- **Policy-modification**

The existing non-exploratory policies can be augmented with an additional mechanism in order to have exploration. This may be achieved by an ε-greedy in which the underlying policy is followed with a high probability $1-\varepsilon$, and a running advertising campaign is chosen at random with a small probability ε. One can derive other possible solutions from the bandit literature, such as the UCB rule [2]. Standing for Upper-Confidence Bound, UCB is a very simple way to achieve asymptotically optimal policy to choose the best action among a set of available actions. Each action is associated to a certain average return; the principle consists in sampling each action, gathering for each its average observed return $\bar{r_i}$, and the number of times each action has been selected n_i. After n actions have been performed, the next action is selected as being the one that maximize the UCB bound: $r_i + \sqrt{\frac{C \ln n}{n_i}}$, where C is an appropriately tuned constant.

- **Estimation-modification**

In this approach, the probability of click estimates are systematically modified (before solving the optimization problem) in order to favor the advertising campaign and user profile couples according to the uncertainty on their estimation based on the following principle: *the more uncertain the estimate, the more exploration may be rewarding*. By giving them artificially a higher probability of click tends to favor their use, and consequently the exploration. For this purpose, [3] use Gittins indices. Similarly one can also use UCB indices associated with the estimates, or with a value sampled from the posterior Beta distribution over the expected reward (see [4]). Empirically, this second way of increasing exploration does not seem to work as well as the first one (for example, ε-greedy with fine-tuned ε) especially if we do not re-plan at each time step. We believe that the reason for this situation is that such methods lead to solutions that only explore the most uncertain areas of the search space.

4 Dynamic setting

Under the static setting of the problem, there are two main constraints: the set of advertising campaigns is known in advance and, consequently, the time horizon is fixed. In the more general and realistic dynamic setting, we remove these constraints. The time horizon is no longer fixed, i.e., does not have a limited length T but instead it is assumed that T is infinite; furthermore, new advertisement campaigns may appear with time. Thus, to the 3 aforementioned categories of advertising campaigns (scheduled, running, and expired), we add a new category made of the yet-unknown campaigns, that is, the advertising campaigns that will come into play in the future, but which future existence is not yet known. In contrast to these unknown advertising campaigns, scheduled, running, and expired advertising campaigns will be qualified as “known”.

In the next two subsections, we will consider two main cases in which either a generative model of advertising campaigns is available, or not. Given a set of parameters and the current state, a generative model generates a stream of advertisement campaigns during a specified time period, together with all related-information, such as, the click probabilities of user profiles for each generated advertising campaign.

4.1 Model based resolution

When a generative model of advertising campaigns is available, it can be utilized to compensate for the uncertainty in future events. In this case, in addition to the set of known advertising campaigns, the model allows us to generate a set of hypothetical unknown advertising campaigns, for example, up to H_{max}[9)], simulating what may happen in future, and include them in the planning phase of NOSEED. By omitting allocations made for these hypothetical advertising campaigns from the (optimal) allocation scheme found by solving the optimization problem, display probabilities that inherently take into consideration the effects of future events can be calculated. Note that this would introduce a bias in the resulting policies which can be reduced by running multiple simulations and combining their results as discussed before.

4.2 Model free resolution

When a generative model is not available, we have an incomplete and uncertain image of the timeline; we only have information about known advertising campaigns, and new advertising campaigns appear periodically or randomly according to a model which is unknown. In this setting, at any time step t, the set of known advertising campaigns (running or scheduled) implies a maximum time horizon H_{max}. Although, it is possible to apply the aforementioned methods and calculate the allocations for the known advertising campaigns, doing so would ignore the possibility of the arrival of new advertising campaigns that may overlap and interact with the existing ones; the resulting long-term policies may perform well if the degree of dynamism in the environment is not high. On the contrary, one can focus only on short or medium-term conditions omitting the scheduled advertising campaigns that start after a not-too-distant time H in the future, i.e., do planning for the advertising campaigns within the chosen planning horizon. The resulting policies will be greedier as H is smaller and disregard the long-time interactions between the existing advertising campaigns; however, they will also be less likely to be affected by the arrival of new campaigns. An example that demonstrates the effect of the planning horizon on the resulting policies is presented in Fig. 6. For such policies, choosing the optimal value of the planning horizon is not trivial due to the fact that it strongly depends on the unknown underlying model. One possible way to overcome this problem would be to solve the problem for a set of different planning horizons $H_1, \ldots, H_u = H_{max}$, and then combine the resulting probability distributions of advertising campaign displays (such as by majority voting).

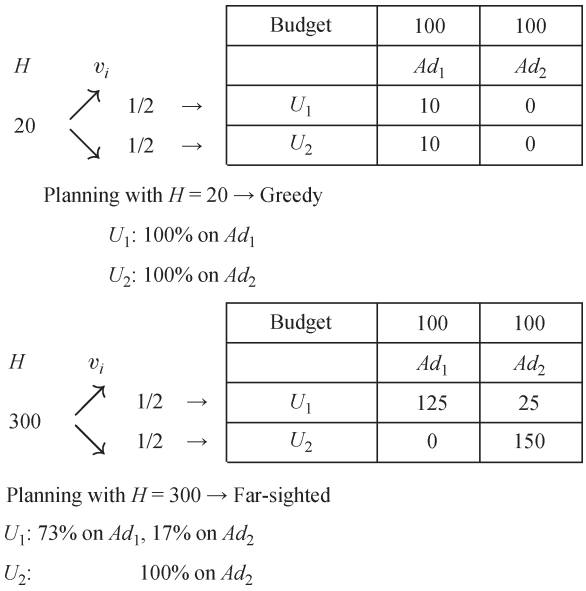

Fig. 6 The effect of the planning horizon H. Ad_1 and Ad_2 start at time 0 and have the same unit profit per click. The click probabilities are $p_{1,1} = 0.8$, $p_{1,2} = 0.1$ for the user profile U_1 and $p_{2,1} = 0.8$, $p_{2,2} = 0.5$ for the user profile U_2. Both profiles have the same visit probability

The sketch of NOSEED algorithm is presented in Fig. 7 and the complete algorithm can be found in the Appendix.

5 Related work

We review the existing work on the problem of advertisement selection for display on web pages, and related problems. We also discuss our own work in respect to these works.

The oldest reference we were able to spot is [5] who mixed a linear program with a simple estimation of CTR to select advertisements to display. In this work, no attention is paid to the exploration/exploitation trade-off and more generally, the problem of the estimation of the CTR is very crudely addressed. Then, [3] introduce a multi-arm bandit approach to balance exploration with exploitation. Their work is based on display proportions, that is unlimited resources; they also

[9)] We will use H to denote the planning horizon and differentiate it from the time horizon of the problem T.

1: Initialize the visit probability estimates for each user profile.
2: **while** there is a request **do**
3: Let U_i be the user profiole of the current visitor
4: **if** there are new advertising campaigns **then**
5: Initialize the click probability estimates of the new campaigns for each user profile.
6: **end if**
7: **if** planning is required **then**
8: Solve the optimization problem and determine the display allocations of advertising campaigns.
9: **end if**
10: Choose an advertising campaign Ad_k basedon U_i and the display allocations of advertising campaigns for the current time interval (with exploration).
11: Display an advertisement from Ad_k to the current visitor and note the outcome (i.e. click or no click).
12: Update visit and click probability estimates according to the outcome.
13: **end while**

Fig. 7 Sketch of the NOSEED algorithm: NOSEED selects advertising campaigns by solving the optimization problem from times to times, and then exploiting its solution to display a certain amount of advertisements

deal with a static set of advertisements. This was later improved by [6] who deal with the important problem of multi-impression of advertisements on a single page; they also deal with the exploration/exploitation trade-off by way of Gittins indices. Ideas drawn from their work on multi-impression may be introduced in ours to deal with that issue.

Aiming at directly optimizing the advertisement selection, side information (information about the type of advertisement, page, date of the request, ...) is used to improve the accuracy of prediction in several recent papers [7–11]. Interestingly, [12] also deals with the multi-impression problem. However, all these works do not consider finite budget constraints, and finite lifetime constraints, as well as the continuous creation of new advertising campaigns; they also do not consider the CTR estimation problem. Recently, [11] focuses on the exploration/exploitation trade-off and proposes interesting ideas that may be combined to ours (varying ε in the ε-greedy strategy, and taking into account the history of the displays of an advertisement). Though not dealing with advertisement selection but news selection, which implies that there is no profit maximization, and no click budget constraint, but merely maximization of the amount of clicks, [13, 14] investigate a multi-arm bandit approach.

Some works have specifically dealt with the accurate prediction of the CTRs, either in a static setting [15], or dealing with a dynamic setting, and non stationary CTRs [16]. [17, 18] also use a hierarchically organized side information on advertisements and pages. Recently, the extent of the content relevance between the pages and the personal interests of users based on intention and sentiment analysis are also considered for improving the predictions [19].

A rather different approach is that of [20] who treated this problem as an on-line bipartite matching problem with daily budget constraints. However, it assumed that we have no knowledge of the sequence of appearance of the profile, whereas in practice we often have a good estimate of it. [21] tried then to take advantage of such estimates while still maintaining a reasonable competitive ratio, in case of inaccurate estimates. Extensions to click budget were discussed in the case of extra estimates about the click probabilities. Nevertheless, the daily maximization of the income is not equivalent to a global maximization.

6 Experiments

We do not see any way to provide a relevant theoretical assessment of this work regarding the performance of the algorithm. Indeed, the algorithm we propose is aimed at dealing with large problems in an efficient way, efficient meaning with the constraint of rather short answering time (“quasi real-time”, that is in the order of the micro-second to decide which advertisement to display). Clearly this constraint on time requires an approximate solution to the problem we consider; however, even if we remove this constraint on time, we are unable to solve exactly the problem we wish to solve within a reasonable amount of time, for a significant size of the problem, so that we can not compare our results with the optimal results. All this makes the experimental assessment a necessity.

Assessing live the approach we propose is impossible; this is a well-known issue of the community. Even if we plugged NOSEED in a real advertisement server, we would have absolutely no way to assess its performance in comparison with an other algorithm. Some workarounds have been proposed

(see e.g., [22][10]), but the issue is clearly not settled today. So, we set up a set of experiments to study its performance, and study how different tunings of the parameters, and how different display policies affect the performance of the algorithm. We report on these experiments in the next sections.

6.1 The generative model

To fit the real-world problem, our approach was tested on a toy-model designed with experts from the research division of Orange Labs. Orange Labs is an important commercial web actor with tens of millions of page views per day over multiple web sites. We took care that each advertising campaign has its own characteristics that more or less appeal to the different visitor profiles.

The model assumes that each advertising campaign Ad_k has a *base click probability* p_k that is sampled from a known distribution (e.g., uniform in an interval, or normally distributed with a certain mean and variance). As clicking on an advertisement is in general a rare event, the base click probabilities are typically low (around 10^{-4}). The click probability of a visitor from a particular user profile is then set to $p_{i,k} = p_k \gamma^{\mathbf{d}-1}$ where $\gamma > 1$ is a predefined multiplicative coefficient, and the random variable $\mathbf{d}$ is sampled from the discrete probability distribution with parameter n that has the following probability mass function $Pr[\mathbf{d} = x] = 2^{n-x}/(2^n - 1), 1 \leqslant x \leqslant n$. When n is small, all advertising campaigns will have similar click probabilities that are close to the base click probability; as n increases, some advertising campaigns will have significantly higher click probabilities for some but not all of the user profiles. Note that, the number of such assignments will be exponentially low; if γ is taken as fixed, then there will be twice as many advertising campaigns with click probability p compared to those with click probability γp. This allows us to effectively model situations in which a small number of advertising campaigns end up being popular in certain user profiles.

In the experiments we used two values for the γ parameter, 2 and 4; experts recommended the use of the latter value, but as we will see shortly, having a higher value for γ may be advantageous for the greedy policy. The value of n is varied between 2 and 6.

6.2 The experiments

Similar to the way that we introduce the proposed method in the previous sections, in the experiments we will also proceed from simpler settings to more complex ones. We opted to focus on core measures and therefore omit some of the extensions that have been discussed in the text.

We begin with the static setting with full information, and uncertainty (Section 2.1.2). In this setting, we consider a fixed time horizon of one day which is assumed to be equivalent to $T = 4 \times 10^6$ page visits. The distribution of user profiles is uniform and the budget and lifetime of advertising campaigns are also sampled uniformly from fixed intervals. In order to determine the starting times of advertising campaigns, we partitioned the time horizon into M equally spaced intervals (in our case 80) and set the starting time of each advertisement campaign to the starting time of an interval chosen randomly, such that the ending times do not exceed the fixed time horizon. The base click probability is set to 10^{-4}. We solved the optimization problem every 10^4 steps.

First, we consider a setting in which there is a single user profile ($N = 1$), and there are $K = 40$ advertising campaigns with an average $L_k = \frac{1}{10}T$, i.e., 1 tenth of the time horizon. All advertising campaigns have the same budget $B_k = B$. Figure 8 shows the relative performance of HLP policy with respect to the HEV policy for different values of the click probability generation parameter n and budgets. We can make two observations: all other parameters being fixed, HLP is more effective with increasing budgets, and the performance gain depends mainly on the value of γ. For $\gamma = 4$, which is

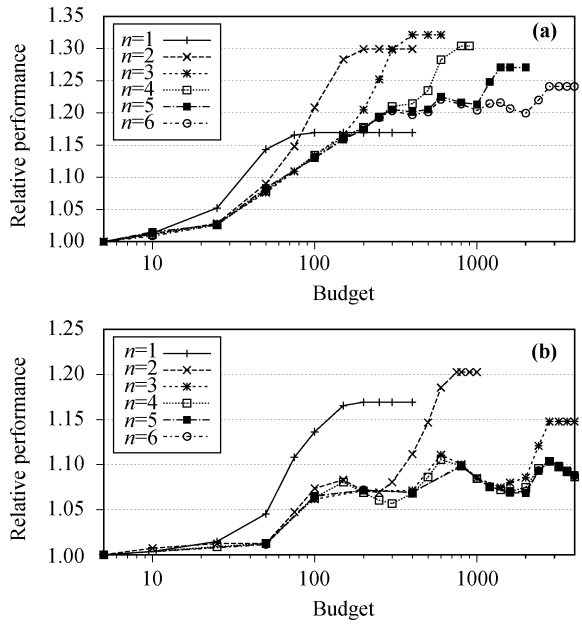

Fig. 8 The relative performance of the HLP policy with respect to the HEV policy for different values of the click probability generation parameter n and budget under the static setting with one user profile and 40 advertising campaigns. The value of γ is either 2 (a) or 4 (b) and the x-axis, i.e., budget B, is in logarithmic scale

[10] Also, Nicol O, Mary J, Preux P. ICML exploration & exploitation challenge: Keep it simple!, 2011, submitted.

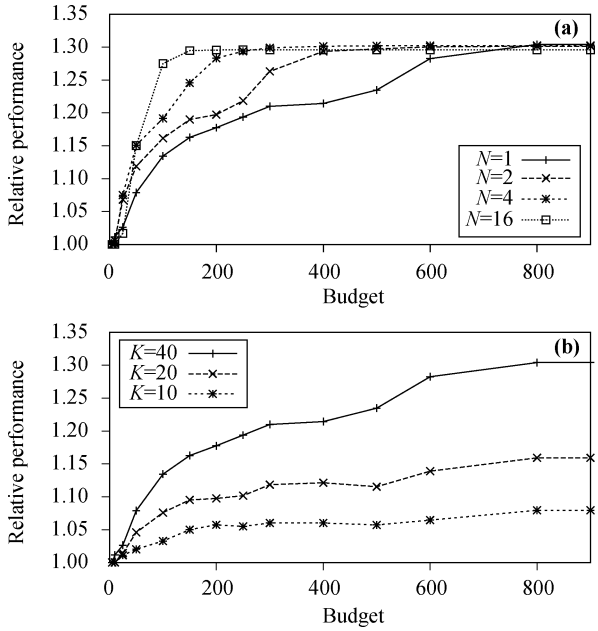

Fig. 9 The effect of the number of (a) user profiles N and (b) advertising campaigns K when other parameters are kept constant and n and γ are set to 2 and 4, respectively

considered to be a realistic value by experts, and reasonable budgets, the greedy policy would perform well. A similar situation also arises when the number of advertising campaigns (K) is low, whereas when the number of user profiles increases, non greedy policies taking longer terms consequences are better (Fig. 9).

Then, in order to isolate and figure out the effect of the overlapping of advertising campaigns on the performance of the algorithms, we conducted another set of experiments in which all advertising campaigns have high click probabilities, and their budget vary depending on their lifetimes. We set:

- n to 1 and sampled the base click probability p_k from a truncated Gaussian distribution with mean 1 and standard deviation 0.02,
- the lifetimes of advertising campaigns L_k are sampled uniformly from 0.5% to 5% of the time horizon,
- the budget B_k of each advertising campaign is set to λ times its lifetime,
- λ is sampled uniformly from the interval $[a, b]$,
- a and b are the parameters of the experiment.

As in the previous case, the time horizon is assumed to be $T = 4 \times 10^6$ page visits. Figure 10 shows the relative performances of HEV, SEV, HLP approaches, as well as the random policy for $K = 100$ advertising campaigns and different values of a and b. We can observe that when the budget to lifetime ratios ($\frac{B_k}{L_k}$) of all campaigns are either low, or high, the difference between the different approaches diminishes. This is due to the fact that in both cases, there is no particular need for taking into account long term consequences: when click probabilities are close to 1, the budget constraints can be satisfied easily when ratios are small (in short, the expected clicks will be grabbed whatever the policy is: no need to be smart), and when they are high choosing any running campaign is likely to end up with a click (in short, whatever the display policy is, however smart it is, budgets can not be fulfilled). However, when the advertising campaigns have diverse budget to lifetime ratios, the interactions between advertising campaigns do matter, and can be exploited by the planning-based approach, especially for low ratios (Fig. 10(a)). In this setting, similar to the toy example presented in Section 3.1.2, the performance of the greedy policy turns out to be inferior to that of random policy which chooses at each time step one of the running advertising campaigns with uniform probability (Fig.10(c)); hence, the stochastic version of the greedy policy, SEV, performs better than HEV (Fig. 10(b)).

Next, we tried longer static settings of over one week period with full, or partial information in which the advertising campaign lifetimes and their budget are more realistic (lifetimes ranging 2 – 5 days, budgets ranging from 500 to 4 000 clicks). The campaigns are generated on a daily basis at the beginning of a simulation, i.e., a set of seven to nine new advertisement are introduced every four million time steps. We tested different values for the click probability generation parameters. There were $N = 8$ user profiles with equal visit probabilities ($v_i = 1/8$). As presented in Fig. 11, in this setting although HLP policy performs better than the greedy policy, the performance gain remains limited. While the greedy policy quickly exploits and consumes new advertisements as they arrive, HLP tends to keep a consistent and uniform click rate at the beginning, and progressively becomes more greedy towards the end of the period (Fig. 12). Figure 13 shows the effect of the planning horizon H: H tunes whether we focus on the campaigns running in the near future (small value of H), or also take into account campaigns that will run in a more remote future (larger value of H). For this experiment, we increased the time horizon from one week to two weeks, and the planning horizon is varied from one day up to the entire time horizon. Note that, the intensity of the interactions between advertising campaigns, in terms of overlapping intervals, and their propagation through time are the main factors that determine the influence of the upcoming campaigns

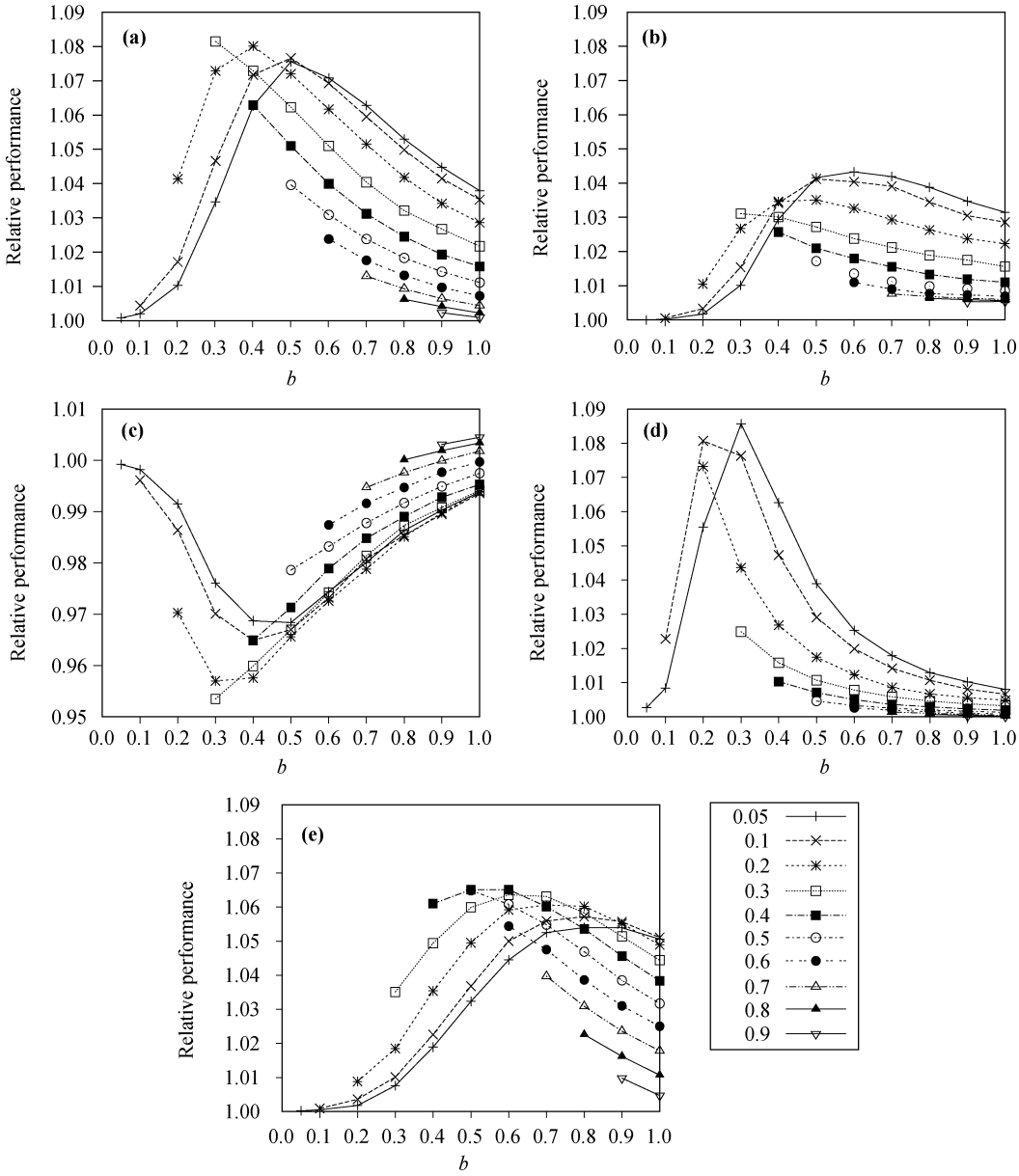

Fig. 10 The relative performances of different approaches for different budget ratios. Each curve presents the case in which the budget of each advertising campaign is set to λ times its lifetime such that λ is sampled uniformly from the interval $[a, b]$, where a is the value that corresponds to the curve and b is the value at the x-axis. (a) HLP vs. HEV, (b) HLP vs. SEV, and (c) HEV vs. random policy with 100 advertising campaigns; HLP vs. HEV with (d) 200 and (e) 50 advertising campaigns

over the display allocations for the currently running campaigns; in this and other experiments we observed that being very far-sighted may not be necessary.

As discussed in Section 3.3.2, when we move to the more realistic setting of partial information, the visits and click probabilities are not known in advance but instead are estimated online. In this setting, without sufficient exploration there is a risk of getting stuck in a local optima; we define local optima as a situation in which the values of some of the options are underestimated and these estimates cannot be improved because the corresponding options are not considered in the search process due to their seemingly low values.

To deal with the exploration-exploitation trade-off, we implemented two approaches: ε-greedy policy, and a UCB based approach. We studied their behavior under various settings in which:

- to increase the variance of the click probabilities, instead of using a fixed value, the base click probabilities p_k of the advertising campaigns are sampled from a Gaussian distribution with mean 0.001 and standard deviation 0.0002,

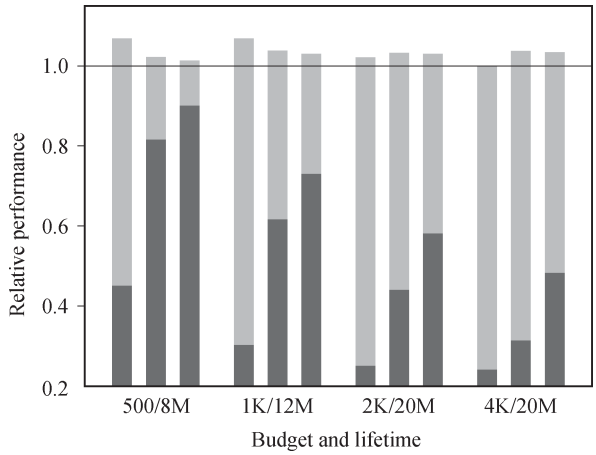

Fig. 11 The performance of the random (dark gray and lowest) and the HLP (light gray and highest) policies with respect to the HES policy under the seven days static setting for different budget (500 to 4 000), lifetime (2 — 5 days) and generation parameter n values. The three sets of bars in each group corresponds to the case where n is taken as to 2, 4, and 6 in that order

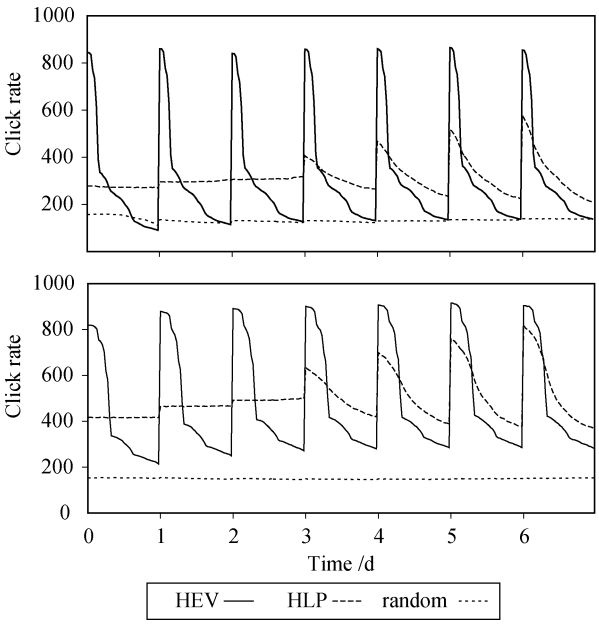

Fig. 12 The moving average of click rate for different policies under the seven day static setting; the lifetime of advertising campaigns is five days and their budgets are either 2 000 (top) or 4 000 (bottom)

- the time horizon is set to $T = 4 \times 10^6$ page visits,
- there are $K = 100$ advertising campaigns,
- their lifetimes L_k falls in the range 0.5% and 5% of the time horizon,
- budget to lifetime ratios $\frac{B_k}{L_k}$ falls in the interval $[0.1, 0.5]$,
- the multiplicative coefficient γ is set to 2.

We employed simple maximum likelihood estimates to estimate click probabilities. Figure 14 shows the results

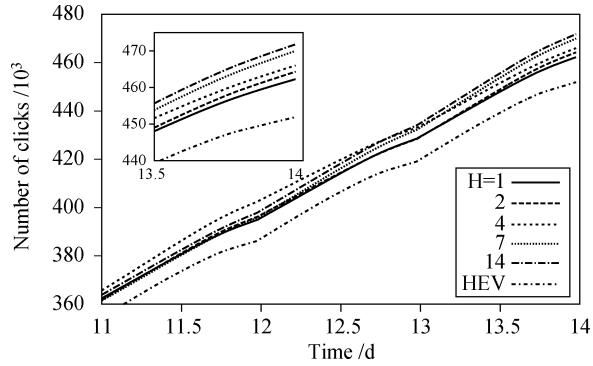

Fig. 13 The effect of horizon H (1, 2, 4, 7 and 14 days) in the 14 days static setting with full information; using less information than available hinders the performance

obtained using both approaches with HEV and HLP policies as a function of ε and the UCB tuning coefficient C. Each subfigure depicts the relative performances of these policies under partial information settings compared to the performance of the HLP policy assuming that the true click probabilities are known (i.e., full information case) for a certain value of click probability generation parameter n, ranging from 1 to 6. The figures highlight that although the performance of ε-greedy varies as a function of ε, for a wide range of values, it performs better than the UCB approach. The performance of the UCB approach is observed to be less sensitive to its tuning coefficient, especially with the HLP policy. It may still be possible that the UCB approach performs better than the ε-greedy approach for a particular value of the tuning coefficient (or a small interval), but fine-tuning the coefficient seems to be more challenging. Furthermore, although ε-greedy has a generally consistent pattern of performance across the full range of n for both HEV and HLP policies, in the UCB approach the performance of the HEV policy deteriorates relative to the HLP policy as n increases, that is, under conditions where the advertising campaigns end up having a wider range of non-homogeneous click probabilities. These results indicate that policy-modification may be a more viable option for balancing the exploitation of the current estimates and exploration.

Finally, we conducted experiments in the dynamic setting with partial information where the probabilities are not known in advance but estimated online. We employed an ε-greedy exploration mechanism with different values of ε and maximum a posteriori estimation with Beta priors. We used the same set of parameters as in two weeks static setting with full information, except that rather than generating all advertising campaigns at the beginning of the simulation, they are generated on a daily basis at the beginning of each day, i.e., a set of seven to nine new advertisement are

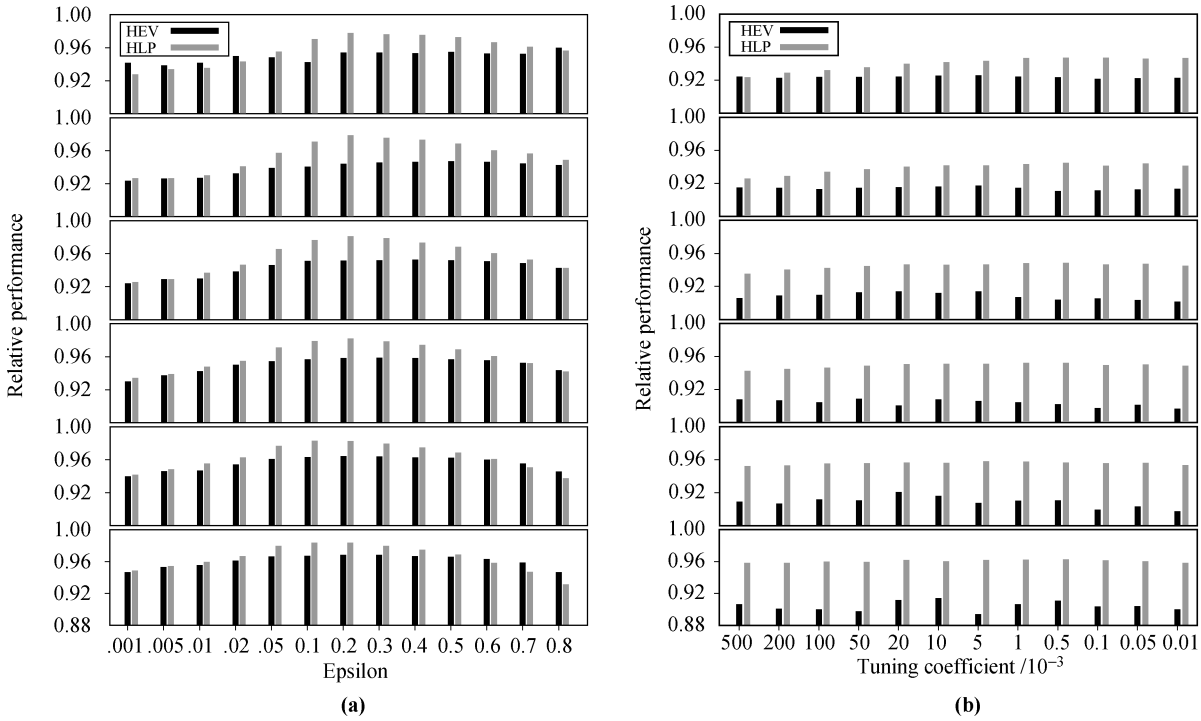

Fig. 14 The performances of HEV and HLP policies with ε-greedy (a) and UCB (b) selection in partial information case relative to the performance of the HLP policy with full information for different values of ε and UCB tuning coefficient; the click probability generation parameter n varies from 1 (top) to 6 (bottom)

introduced every 4 million time steps; the planning is done over all known advertising campaigns. The results presented in Fig. 15 show that HLP can perform better than HEV; however for both policies, the chosen set of hyper-parameters influences the outcome.

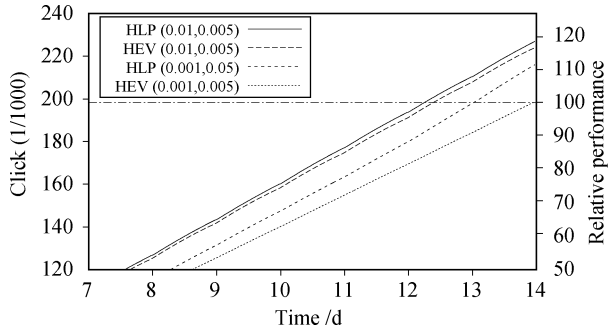

Fig. 15 The performance of HEV and HLP algorithms in the dynamic setting with partial information using ε-greedy exploration. The numbers in parentheses denote the values of the hyper-parameters of the Beta prior (α and β parameters are set to be equal to each other) and ε

7 Conclusion and future work

In this paper, we considered the advertisement selection problem for display on web pages. Aiming at considering the problem in the most realistic setting, and providing effective and efficient algorithms to perform this selection on a production system, we have formalized the problem by providing a series of increasing complexity settings. This let us discuss various algorithmic approaches, and clearly identify the issues. While defining this set of problems, we provided a way to effectively tackle this problem, and provided an experimental study of some of their key features. The experimental study is based on a realistic model, carefully designed with a major commercial Internet portal.

We have shown that optimizing advertisement display handling finite budgets and finite lifetimes in a dynamic and non stationary setting, is feasible within realistic computational time constraints. We have also given some insights in what can be gained by handling this constraint, depending on the properties of the advertisements to display. We have also exhibited that lifetime of the advertisements impact the overall performance, and so should be taken into account into the pricing policy. Moreover our work may be seen as a part of a decision aid tool. For instance, it can help to price the advertisements in the case in which a fraction of the advertising campaigns are in the "cost per display" model, while the rest is in the cost per click model. This is rather easy because the LP solution provides an estimation of the profit for each visitor profile.

Our work shows that depending on the parameters and characteristics of the existing or prospective advertising cam-

paigns, a simple greedy approach may perform well or one can benefit from using a more advanced solution, such as NOSEED, that takes into consideration the long term gains. Figure 8 illustrates how these parameters interact. To summarize, we may say that if there are few overlapping advertisements, or many advertisements with long lifetimes and good click rates, then we should be greedy. Between these two extreme solutions, one should consider the constraints associated to each advertising campaign.

This work calls for many further developments. A possibility is to solve the problem from the perspective of the advertiser, i.e., help the advertiser to set the value of a click, and adjust it optimally with respect to his/her expected number of visitors. It would be equivalent to a local sensitivity analysis of the LP problem. A more difficult issue is that of handling multiple advertisement displays on the same page. It may be possible to handle them by estimating the correlation between the advertisements, and trying to update multiple click probabilities at the same time. Some recent developments in the bandit setting [23] are interesting in this regard.

We are willing to draw some theoretical results on how far from the optimal strategy we are. Dealing with finite resources, under finite time constraints, in a dynamic setting makes that kind of study very difficult. An other work originates from the analysis of some real web server logs. We have already been very slightly using such source of information, but much more has to be done. We also think that it is important to go towards learning on-line the profiles of the visitors depending on their click behavior instead of having pre-existing ones.

Acknowledgements This research was supported, and partially funded by Orange Labs, under externalized research contract number CRE number 46 146 063 – 8360, and by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council and FEDER through the "Contrat de Plan État-Région (CPER) 2007-2013", and the contract "Vendeur Virtuel Ubiquitaire" of the "Pôle de compétitivité Industries du Commerce".

The model used in this paper, as well as its parameters have been designed with, and validated by, Orange Labs, to keep up with the essential characteristics of the real problem.

The realization of simulations were carried out using the Grid'5000 experimental testbed, an initiative from the French Ministry of Research, INRIA, CNRS and RENATER and other contributing partners.

Appendixes

Notations

	Indices
t	Current time
i	Index of a user profile
k	Index of an advertising campaign
j	Index of a time interval
	User profile
N	# user profiles (Sec. 2)
U_i	User profile i (Sec. 2)
v_i	Probability that a certain visitor belongs to U_i (Sec. 2)
	Advertising campaign
C	The set of known (running) advertising campaigns at a given time (Sec. 3.1.2)
K	# of advertising campaigns (Sec. 3.1)
Ad_k	Advertising campaign k (Sec. 2)
S_k	Starting time of Ad_k (Sec. @2)
L_k	Lifetime of Ad_k (Sec. 2)
B_k	Budget of Ad_k (Sec. 2)
cp_k	Click profit of Ad_k (Sec. 2)
rb_k	$\leqslant B_k$: remaining budget of Ad_k (Sec. 2)
E_k	Ending time of Ad_k ($E_k = S_k + L_k$, Sec. 3.1.3)
$p_{i,k}$	Probability that a visitor $\in U_i$ clicks on an advertisement $\in Ad_k$ (Sec. 2)

	Time
T	Time horizon of the problem (Sec. 2.1.1.1)
H	Time horizon of the resolution (planning, Sec. 2.2.1)
M	# of time intervals (Sec. 3.1.3)
I_j	Time interval j (Sec. 3.1.3)
AI_j	The set of running advertising campaigns in I_j (Sec. 3.1.3)
IA_k	The set of time interval is which Ad_k is running (Sec. 3.1.3)
$visit_i$	# of visit $\in U_i$ for time $\leqslant t$ (Sec. 3.3.1)
$click_{i,k}$	# of times a user $\in U_i$ clicked on Ad_k, for time $\leqslant t$ (Sec. 3.3.1)
$display_{i,k}$	# of times an Ad_k has been displayed to a visitor $\in U_i$ for time $\leqslant t$ (Sec. 3.3.1)
	Display allocation (Sec. 3.1.3)
$a_{i,k,j}$	# of advertisement displays allocated to Ad_k in interval I_j for user profile U_i
$\hat{a}_{i,k,j}$	$= a_{i,k,j} / \sum_{Ad_k \in AI_j} a_{i,k,j}$ is the ratio of allocation of displays for U_i and Ad_k during I_j to the total allocations for that user profile in the same interval (forming a categorical distribution)
	Exploration policy parameters (Sec. 3.3.2)
ε	Parameter of the ε greedy-policy
C	Parameter of the UCB policy
	Miscellaneous
s	A state of the MDP (Sec. 3.1.1)
ϑ_i	Prior for the maximum likelihood estimation of v_i
λ (λ_k, $\lambda_{i,j}$)	Parameters of the Poisson distribution (its interpretation is a risk ratio, Sec. 3.2)
Λ_k	Risk ratio threshold (Sec. 3.2)
$\alpha_{i,k}, \beta_{i,k}$	Parameters of the Beta distribution (Sec. 3.3.1)
p_k	Base click probability for advertisements $\in Ad_k$ (Sec. 6.1)
$\gamma, n, [a, b]$	Parameters of the generative model (Sec. 6.1 and 6.2)

The NOSEED algorithm

Input: N: number of user profiles; T: time horizon; H: planning horizon; C: set of known advertising campaigns; hyper-parameters of click and visit probability estimators, (e.g. $\alpha_{i,k}, \beta_{i,k}$ for the Beta distributions).

Additional variables: C_{last}: set of known advertising campaigns at last planning; $\hat{p}_{i,k}$: probability distribution for the estimate of $p_{i,k}$ (a Beta distribution).

procedure CheckForNewCampaigns
  for all $Ad_k \in C$ and $Ad_k \notin C_{last}$ /* *New campaigns* */ **do**
    for $i = 1$ **to** N **do**
      $\hat{p}_{i,k} = Beta(\alpha_{i,k}, \beta_{i,k})$ /* *Initial click probability estimates* */
      $click_{i,k} = display_{i,k} = 0$
    end for
  end for
end procedure

/*C: *set of advertising campaigns; rb: remaining budgets of advertising campaigns in* C, rb_k *denotes the remaining budget of* Ad_k */
function FindAllocations (t, C, rb)
  $boundaries = \{min(t, S_k)\} \cup \{E_k\}, \forall Ad_k \in C$ such that $S_k \leqslant t + H$.
  Let $t_0, \ldots, t_M$ is the sorted list of the elements of $boundaries$ and define intervals $I_j = [t_{j-1}, t_j], 1 \leqslant j \leqslant M$.
  for $j = 1$ **to** M **do**

$l_j = t_j - t_{j-1}$ /*length of the interval */
$AI_j = \{Ad_k | S_k < t_j \leqslant E_k\}$ /*the set of campaigns in each interval */
end for
for all $Ad_k \in C$ **do**
$IA_k = \{I_j | Ad_k \in AI_j\}$ /*the set of intervals that cover Ad_k */
end for
$\mathcal{A} = \{I_j | AI_j \neq \emptyset\}$ /*non-empty intervals */
Let $a_{i,k,j}$ denote the number of displays allocated to the campaign Ad_k in interval I_j for the user profile U_i.
Solve the linear program maximize $\sum_{I_j \in \mathcal{A}} \sum_{Ad_k \in AI_j} cp_k p_{i,k} a_{i,k,j}$ with the set of constraints
(a) $\sum_{Ad_k \in AI_j} a_{i,k,j} \leqslant v_i l_j, \forall\, U_i, I_j \in \mathcal{A}$
(b) $\sum_{U_i} \sum_{I_j \in IA_k} p_{i,k} a_{i,k,j} \leqslant rb_k, \forall\, Ad_k \in C$
(c) $\sum_{U_i} \sum_{Ad_k \in AI_j} a_{i,k,j} \leqslant l_j, \forall I_j \in \mathcal{A}$
Let *intervals* be the list of intervals I_j and *allocations* be the list of display allocations $a_{i,k,j}$.
return [*intervals*, *allocations*]
end function

function DoPlanning (t)
$C_{last} = C$ /*Save the current set of advertising campaigns */
if a generative model is available **then**
Generate a set of hypothetical campaigns C'. /* up to time $t + H$ */
$C_{current} = C \cup C'$
else
$C_{current} = C$
end if
Update click probability estimates, i.e., $\hat{p}_{i,k} = Beta(\alpha_{i,k} + click_{i,k}, \beta_{i,k} + display_{i,k})$
if using estimation-modification approach
Modify $\hat{p}_{i,k}$, e.g. using Gittins or UCB indices /* see Section 3.3.2 */
end if
for all $Ad_k \in C_{current}$ **do**
if risk factor $\Lambda_k < 1$ **them** /*Modify budget limits for dealing with uncertainty, see Section 3.2 */
$rb'_k = \arg\min_\lambda Pr(Po(\lambda) > rb_k - 1) \geqslant \Lambda_k$
else
$rb'_k = rb_k$
end if
end for
return FindAllocations $(t, C_{current}, rb')$
end function

/* U_i is the profile of a visitor. intervals and allocations are the list of intervals and display allocations in each interval as determined in the planning phase, i.e. DoPlanning function, respectively; I_j denotes the j^{th} interval, and $a_{i,j,k}$ denotes the number of advertisement displays allocated to $Ad_k \in I_j$ for U_i. */
function ChooseCampaign $t, U_i, intervals, allocations$
Determine $I_j = [t_{j-1}, t_j] \in intervals$ such that $t_{j-1} \leqslant t < t_j$ /* current interval */
Let $AI_j \subseteq C$ be the set of running campaigns that span I_j.
if $AI_j = \emptyset$ **then**
return $\emptyset$ /* There is no running advertising campaign */
end if
$\bar{a}_{i,j} = \sum_{Ad_k \in AI_j} a_{i,k,j}$ /* Total allocations in this interval */

```
61:     if $\overline{a}_{i,j} > 0$ then
62:       for all $Ad_k \in AI_j$ do
63:         $\hat{a}_{i,k,j} = a_{i,k,j}/\overline{a}_{i,j}$ /* Calculate display probabilities */
64:       end for
65:       Choose an advertising campaign $Ad_k$ based on $\hat{a}_{i,k,j}$ /* e.g. using HLP or SLP with exploration, if any */
66:       $a_{i,k,j} = a_{i,k,j} - 1$ /* Update the allocation for $Ad_k$ */
67:     else
68:       Choose a campaign $Ad_k$ from $AI_j$ (e.g. randomly). /* No allocations to display for this user profile */
69:     end if
70:     return $k$
71: end function
72:
73: /* The main loop */
74: $C_{last} = \emptyset$
75: for $i = 1$ to $N$ do /* Initialize visit probability estimates */
76:     $visit_i = 1, v_i = 1/N$ /* $\vartheta_i = 1$ */
77: end for
78: $t = 0$ /* Set time to 0 */
79: while there is a request do
80:     Let $U_i$ be the user profile of the current visitor.
81:     CheckForNewCampaigns
82:     if $t = 0$ or planning is required /* e.g. when an advertising campaign expires or periodically */ then
83:         $[intervals, allocations]$ =DoPlanning $(t)$
84:     end if
85:     $k$ =ChooseCampaign $(t, U_i, intervals, allocations)$
86:     if $Ad_k \neq \emptyset$ then
87:       $display_{i,k} = display_{i,k} + 1$
88:       if visitor clicks on $Ad_k$ then
89:         $click_{i,k} = click_{i,k} + 1$ /* Update the click count of the user profile */
90:         $rb_k = rb_k - 1$ /*Update the remaining budget of the advertising campaign */
91:       end if
92:     end if
93:     $t = t + 1$
94:     /* Update the visit probability estimates */
95:     $visit_i = visit_i + 1$
96:     for $i = 1$ to $N$ do
97:       $v_i = visit_i/(t + N)$
98:     end for
99: end while
```

References

1. Puterman M L. Markov Decision Processes: Discrete Stochastic Dynamic Programming. New York: John Wiley & Sons, 1994
2. Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 2002, 47(2-3): 235–256
3. Abe N, Nakamura A. Learning to optimally schedule Internet banner advertisements. In: Proceedings of the 16th International Conference on Machine Learning. 1999, 12–21
4. Granmo O C. A Bayesian learning automaton for solving two-armed Bernoulli bandit problems. In: Proceedings of the 7th International Conference on Machine Learning and Applications. 2008, 23–30
5. Langheinrich M, Nakamura A, Abe N, Kamba T, Koseki Y. Unintrusive customization techniques for web advertising. Computer Networks, 1999, 31(11-16): 1259–1272
6. Nakamura A, Abe N. Improvements to the linear programming based scheduling of web advertisements. Electronic Commerce Research,

2005, 5(1): 75–98

7. Pandey S, Agarwal D, Chakrabarti D, Josifovski V. Bandits for taxonomies: a model-based approach. In: Proceedings of the 7th SIAM International Conference on Data Mining. 2007
8. Langford J, Zhang T. The epoch-greedy algorithm for multi-armed bandits with side information. In: Proceedings of 20th Advances in Neural Information Processing Systems. 2008, 817–824
9. Wang C C, S.R. Kulkarni S R, Poor H V. Bandit problems with side observations. IEEE Transactions on Automatic Control, 2005, 50(3): 338–355
10. Kakade S M, Shalev-Shwartz S, Tewari A. Efficient bandit algorithms for online multiclass prediction. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 440–447
11. Li W, Wang X, Zhang R, Cui Y, Mao J, Jin R. Exploitation and exploration in a performance based contextual advertising system. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2010, 27–36
12. Pandey S, Olston C. Handling advertisements of unknown quality in search advertising. In: Proceedings of 18th Advances in Neural Information Processing Systems. 2006, 1065–1072
13. Agarwal D, Chen B, Elango P. Explore/exploit schemes for web content optimization. In: Proceedings of the 9th IEEE International Conference on Data Mining. 2009, 1–10
14. Li L, Chu W, Langford J, Schapire R E. A contextual-bandit approach to personalized article recommendation. In: Proceedings of the 19th International Conference on World Wide Web. 2010, 661–670
15. Richardson M, Dominowska E, Ragno R. Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th International Conference on World Wide Web. 2007, 521–530
16. Agarwal D, Chen B C, Elango P. Spatio-temporal models for estimating click-through rate. In: Proceedings of the 18th International Conference on World Wide Web. 2009, 21–30
17. Agarwal D, Broder A, Chakrabarti D, Diklic D, Josifovski V, Sayyadian M. Estimating rates of rare events at multiple resolutions. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2007, 16–25
18. Wang X, Li W, Cui Y, Zhang B, Mao J. Clickthrough rate estimation for rare events in online advertising. In: Hua X S, Mei T, Hanjalic A, eds. Online Multimedia Advertising: Techniques and Technologies. Hershey: IGI Global, 2010
19. Fan T K, Chang C H. Sentiment-oriented contextual advertising. Knowledge and Information Systems, 2010, 23(3): 321–344
20. Mehta A, Saberi A, Vazirani U, Vazirani V. Adwords and generalized on-line matching. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science. 2005, 264–273
21. Mahdian M, Nazerzadeh H. Allocating online advertisement space with unreliable estimates. In: Proceedings of the 8th ACM Conference on Electronic Commerce. 2007, 288–294
22. Langford J, Strehl A, Wortman J. Exploration scavenging. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 528–535
23. Koolen W M, Warmuth M K, Kivinen J. Hedging structured concepts. In: Proceedings of the 23rd Annual Conference on Learning Theory. 2010, 93–105

Sertan Girgin has two BSc degrees, one in Computer Engineering and the other in Mathematics, and a PhD degree in Computer Engineering from Middle East Technical University (METU), Turkey, 2007. He was a visiting researcher at the Department of Computer Science, University of Calgary, Canada, in 2006. For three years, Dr. Girgin worked as a postdoc researcher in team-project Sequel, INRIA Lille Nord Europe, France. Currently, he is with Google, Inc. His research interests include sequential learning, evolutionary computation, distributed AI and multi-agent systems.

Jérémie Mary is Assistant professor at University of Lille and member of the SequeL team at INRIA. He is also member of the european network of excellence PASCAL 2. He obtained his PhD on online machine learning, at Université Paris XI advised by Michèle Sebag and Antoine Cornuéjols. His mains interests in research are related to Machine Learning and more specifically sequential data. With Olivier Nicol (PhD student), he won the ICML'2011 challenge Exploration&Exploitation on data provided by Adobe.

Philippe Preux defended his PhD in computer science in 1991, at the Université de Lille, France. He is currently professor in computer science at the Université de Lille. He is the head of the SequeL research group, affiliated to both INRIA, CNRS, and the university. Since 1991, his research focuses on adaptive systems. He has worked on genetic algorithms and metaheuristics for combinatorial optimization; he then moved to reinforcement learning. These days, his main research interests are statistical learning on sequential data, data mining and sequential decision making in face of very large amounts of data, in non stationary environments.

Olivier Nicol holds a Master's degree in computer science with specialization in software engineering from the University of Lille, France. He is now studying for a PhD under Philippe Preux and Jérémie Mary in the SequeL (Sequential Learning) team at INRIA Lille. His main research interests lie in

Machine learning and especially using sequential data such as web logs. For instance he is currently working on how to use data to evaluate recommendation policies (and more generally contextual bndits policies) without having to actually test them on the real world. Together with Jérémy Mary he won the ICML 2011 Exploration and Exploitation challenge which was about balancing exploration and exploitation in order to efficiently recommend items to visitors on an Abode web site.