2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Online Advertisement, Optimization and Stochastic Networks

Bo (Rambo) Tan and R. Srikant
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL, USA

***Abstract*— In this paper, we propose a stochastic model to describe how search service providers charge client companies based on users' queries for the keywords related to these companies' ads by using certain advertisement assignment strategies. We formulate an optimization problem to maximize the long-term average revenue for the service provider under each client's long-term average budget constraint, and design an online algorithm which captures the stochastic properties of users' queries and click-through behaviors. We solve the optimization problem by making connections to scheduling problems in wireless networks, queueing theory and stochastic networks. Unlike prior models, we do not assume that the number of query arrivals is known. Due to the stochastic nature of the arrival process considered here, either temporary "free" service, i.e., service above the specified budget (which we call "overdraft") or under-utilization of the budget (which we call "underdraft") is unavoidable. We prove that our online algorithm can achieve a revenue that is within $O(\epsilon)$ of the optimal revenue while ensuring that the overdraft or underdraft is $O(1/\epsilon)$, where ϵ can be arbitrarily small. With a view towards practice, we also show that one can always operate strictly under the budget.**

Our algorithm also allows us to quantify the effect of errors in click-through rate estimation on the achieved revenue. We show that we lose at most $\frac{\Delta}{1+\Delta}$ fraction of the revenue if Δ is the relative error in click-through rate estimation.

I. Introduction

Providing online advertising services has been the major source of revenue for search service providers such as Google, Yahoo and Microsoft. When an Internet user queries a keyword, alongside the search results, the search engine may also display advertisements from some companies which provide services or goods related to this keyword. These companies pay the search service providers for posting their ads with a specified amount of price for each ad on a pay-per-impression or pay-per-click basis. We call them "clients" in the following text.

Maximizing the revenue obtained from their clients is the key objective of search service providers. Research which targets this objective has followed two major directions. One is based on auction theory, in which the goal is to design mechanisms in favour of the service provider, and much of the research in this direction considers static bids (e.g. [8]; see [6] for a survey), while dynamic models such the one in [15] are still emerging. The other is from the perspective of online resource allocation without considering the impact of the service provider's mechanisms on the clients' bids, and the main focus of this kind of research is on designing an online algorithm which posts specific ads in response to each search query arriving online, in order to achieve a high competitive ratio with respect to the offline optimal revenue. Our work follows the second direction.

Our model is as follows:

Online Advertising Model:

Assume that queries for keyword q arrive to the search engine according to a stochastic process at rate ν_q queries per time slot, where we have assumed that time is discrete and a *"time slot"* is our smallest discrete time unit. In response to each query arrival, the search engine may display ads from some clients on the webpage. There may be a number of places (e.g., top, bottom, left, right, etc.) on a webpage where ads could be displayed. We will call these places *"webpage slots."* When client i's ad is displayed in webpage slot s when keyword q is queried, there is a probability with which the user who is viewing the page (the one who generated the query) will click on the ad. This probability, called the *"click-through rate,"* is denoted by c_{qis}.

A client specifies the amount of money ("bid") that it is willing to pay to the search service provider when a user clicks on its ad related to a specific query. We use r_{qi} to denote this per-click payment from client i for its ad related to a query for keyword q. Additionally, client i also specifies an average budget b_i which is the maximum amount that it is willing to pay per *"budgeting cycle"* on average, where a budgeting cycle equals to N time slots (we have introduced the notion of a budgeting cycle since the time-scale over which queries arrive may be different than the time-scales over which budgets may be settled).

The problem faced by the search service provider is then to assign advertisements to webpage slots, in response to each query, so that its long-term average revenue is maximized.

Based on the above model, we design an online algorithm which achieves a long-term average revenue within $O(\epsilon)$ of the offline optimal revenue, where ϵ can be chosen arbitrarily small, indicating the near-optimality of our online algorithm. Before entering into the details, in the next two subsections we will first survey the related literature, highlight the main contributions of our work, and discuss the differences between our model and previous ones.

A. Related Work

We will only survey the online resource allocation models here, and not the auction models.

Mehta et al. [14] modeled the online ads problem as a generalization of an online matching problem [11] on a bipartite graph of queries and clients. Later in [2], Buchbinder et al. showed that matching clients to webpage slots (whether it is a single slot or multiple slots) can be solved as a maximum-weighted matching problem. Following [2], a number of other online algorithms using the maximum-weighted bipartite matching idea have been proposed in [13], [5], [3] and [4] (although earlier than [2], essentially [10] and [14] were also using this maximum-weight matching idea).

The algorithms in [2], [5], [10] and [14] are $1 - 1/e$ competitive. By modifying the algorithm in [14], Mahdian et al. [13] designed a class of algorithms which achieve a considerably better competitive ratio with accurate estimates of the number of query arrivals for each keyword, while still guarantee a reasonably good competitive ratio with inaccurate estimates. Two learning-based algorithms in [3] and [4] achieve a near-optimal competitive ratio of $1-O(\epsilon)$ based on a random-order arrival model (rather than the adversarial model in most of the earlier work), assuming small bids and knowledge of the total number of queries.

All of the algorithms in [2], [5], [3] and [4] use a primal-dual framework to compute a maximum-weighted matching at each iteration, in which the dual variables (corresponding to each client) are used to determine the weights. Their difference is that the algorithms in [2] and [5] dynamically update the dual variables, while those in [3] and [4] take the first ϵ fraction of queries to learn the optimal dual variables (with respect to this training set) and kept static for use in following queries.

A detailed survey can be found in our technical report [16].

B. Our Contributions and Comparison to Prior Work

As in prior work (especially [2] and [5]), our solution relies on a primal-dual framework to solve a maximum-weighted matching problem on a bipartite graph of clients and webpage slots, with dynamically updated dual variables which contribute to the weights on the edges of the bipartite graph. However, unlike prior work, we are able to obtain a revenue which is $O(\epsilon)$ close to the optimal revenue using a purely adaptive algorithm without the need for the knowledge of the number of query arrivals over a time period or the average arrival rates.

Our solution is related to scheduling problems in wireless networks. In particular, we use the optimization decomposition ideas in [7], [9] and the stochastic performance bounds in [12]. Borrowing from that literature, we introduce the concept of an *"overdraft" queue*. The overdraft queue measures the amount by which the provided service temporarily exceeds the budget specified by a client.

Our online algorithm exhibits a trade-off between the revenue obtained by the service provider and the level of overdrafts. Specifically, our algorithm can achieve a revenue that is within $O(\epsilon)$ of the optimal revenue while ensuring that the overdraft is $O(1/\epsilon)$, where ϵ can be arbitrarily small. We can further modify our online algorithm so that clients can always operate strictly under their budgets. Finally, our algorithm and analysis naturally allow us to assess the impact of click-through rate estimation on the service providers revenue.

There are two points of departure in our algorithm compared to existing models: the first one is that we assume a purely stochastic model in which the query arrival rates are unknown. Thus, there is no need to know the number of arrivals in a time period as in prior models. The other is that we assume an average budget rather a fixed budget over a time horizon. This allows us to better model permanent clients (e.g., big companies who do not stop advertising) and who do not provide a fixed time-horizon budget. Clients who advertise for a limited amount of time are not explicitly modeled here. But we believe that our model will also handle such clients well since the algorithm is naturally adaptive. For such clients, one can divide their total budget by the time horizon over which they contract with the service provider to approximately model them in our framework.

A minor difference with respect to prior models is that our model assumes that time is slotted. This can be easily modified to assume that query arrivals can occur at any time according to some continuous-time stochastic process. The only difference is that our analysis would then involve continuous-time Lyapunov drift instead of the discrete-time drift used in this paper. From a theoretical point of view, our analysis is different from prior work which uses competitive ratios: our model and solution is similar in spirit to stochastic approximation [1] where gradients (here the gradient of the dual objective) are known only with stochastic perturbations. This point of view is essential to model stochastic traffic with unknown statistics.

Instead of the $1-O(\epsilon)$ competitive ratio in prior work, we show that our algorithm achieves a revenue which is within $O(\epsilon)$ of the optimal revenue. The $O(\epsilon)$ penalty arises due to the stochastic nature of our model. However, we do not require assumptions such as knowledge of the total number of queries in a given period [13], [3], [4], or information of keyword frequencies [13].[1]

C. Organization of the Paper

The rest of the paper is organized as follows: In Section II, we formulate an optimization problem involving long-term averages. In Section III, we start considering the stochastic version of our model and propose an online algorithm, which also introduces the concept of "overdraft queue." Performance analysis of this online algorithm, which includes the near-optimality of the long-term revenue and an upper bound on the overdraft level, will also be done in Section III. Section IV gives two extensions, namely the "underdraft" mechanism and the decisions based on estimated click-through rates. Section V concludes the whole paper.

II. An Optimization Problem Involving Long-Term Averages

Based on the model described in Section I, we first pose the revenue maximization problem as an optimization problem involving long-term averages. For this purpose, we

[1] It should be mentioned that another common assumption "small bids" (or "large budgets", "large offline optimal value") used in [10], [14], [13], [5], [3] and [4] is not essentially different from our "long-term" assumption.

define an assignment of clients to webpage slots as a matrix M of which the $(i,s)^{\text{th}}$ element is defined as follows:

$$M_{is} = \begin{cases} 1, & \text{if client } i \text{ is assigned to webpage slot } s \\ 0, & else. \end{cases}$$

The matrix M has to satisfy some practical constraints. First, a webpage slot can be assigned to only one client and vise versa. Furthermore, the assignment of clients to certain webpage slots may be prohibited for certain queries. For example, it may not make sense to advertise chocolates when someone is searching for information about treatments for diabetes. These constraints can be abstracted as follows: For the queried keyword q, the set of assignment matrices have to belong to some set $\mathcal{M}_q$. We also let p_{qM} be the probability of choosing matrix M when the queried keyword is q.

The optimization problem is then given by

$$\max_{\mathbf{p}} \sum_q \nu_q \sum_{M\in\mathcal{M}_q} p_{qM} \sum_{i,s} M_{is} c_{qis} r_{qi} \tag{1}$$

subject to

$$N \sum_q \nu_q \sum_{M\in\mathcal{M}_q} p_{qM} \sum_s M_{is} c_{qis} r_{qi} \le b_i, \qquad \forall i; \tag{2}$$

$$0 \le p_{qM} \le 1, \qquad \forall q,\ M \in \mathcal{M}_q; \tag{3}$$

$$\sum_{M\in\mathcal{M}_q} p_{qM} \le 1, \qquad \forall q. \tag{4}$$

In the above formulation, the objective (1) is the average revenue per time slot and constraint (2) expresses the fact that the average payment over a budgeting cycle should not exceed the average budget. The optimization is a linear program and if all the problem parameters are known, in principle, it can be solved offline, returning probabilities $\{p_{qM}\}$ which can be used by a service provider to maximize its revenue. However, such an offline solution is not desirable for at least two reasons:

- Being a static approach, it does not use any feedback about the current state of the system. For example, the fact that the empirical average payment of a client has severely exceeded its average budget would have no impact on the subsequent assignment strategy. Since the formulation and hence, the solution, only cares about long-term budget constraint satisfaction, severe overdraft or underdraft of the budget can occur over long periods of time.
- The offline solution is a function of the query arrival rates $\{\nu_q\}$. Thus, a change in the arrival rates would require a recomputation of the solution.

In view of these limitations of the offline solution, we propose an online solution which adaptively assigns client advertisements to webpage slots to maximize the revenue. As we will see, the online solution does use feedback about the overdraft (or underdraft) level in future decisions, and does not require knowledge of $\{\nu_q\}$.

III. Online Algorithm and Performance Analysis

A. A Dual Gradient Descent Solution

To get some insight into a possible adaptive solution to the problem, we first perform a dual decomposition which suggests a gradient solution. However, a direct gradient solution will not take into the account the stochastic nature of the problem and will also require knowledge of the query arrival rates $\{\nu_q\}$. We will address these issues in the following subsections, using techniques that, to the best of our knowledge, have not been used in prior literature on the online advertising problem.

We append the constraint (2) to the objective (1) using Lagrange multipliers $\delta_i \ge 0$ to obtain a partial Lagrangian function

$$\begin{aligned} L(\mathbf{p},\boldsymbol{\delta}) = &\sum_q \nu_q \sum_{M\in\mathcal{M}_q} p_{qM} \sum_{i,s} M_{is} c_{qis} r_{qi} - \sum_i \delta_i \cdot \\ &\left(\sum_q \nu_q \sum_{M\in\mathcal{M}_q} p_{qM} \sum_s M_{is} c_{qis} r_i - \frac{b_i}{N} \right) \\ = &\sum_q \nu_q \sum_{M\in\mathcal{M}_q} p_{qM} \sum_{i,s} M_{is} c_{qis} r_{qi} (1-\delta_i) + \sum_i \frac{\delta_i b_i}{N}, \end{aligned}$$

subject to constraints (3) and (4). The dual function is

$$D(\boldsymbol{\delta}) = \max_{\mathbf{p}} \sum_q \nu_q \sum_{M\in\mathcal{M}_q} p_{qM} \sum_{i,s} M_{is} c_{qis} r_{qi} (1-\delta_i) + \sum_i \frac{\delta_i b_i}{N},$$

subject to constraints (3) and (4). Note that the maximization part in the dual function can be decomposed into independent maximization problems with regard to each queried keyword q, i.e., for all q,

$$\begin{aligned} &\max_{\{p_{qM},\ M\in\mathcal{M}_q\}} \sum_{M\in\mathcal{M}_q} p_{qM} \sum_{i,s} M_{is} c_{qis} r_{qi} (1-\delta_i) \\ = &\max_{M\in\mathcal{M}_q} \sum_{i,s} M_{is} c_{qis} r_{qi} (1-\delta_i), \end{aligned}$$

where it is easy to see that each maximization is solved by a deterministic solution. This suggests the following primal-dual algorithm to iteratively solve the original optimization problem (1): at step k,

$$\forall q, \qquad \hat{M}^*(q,k) \in \arg\max_{M\in\mathcal{M}_q} \sum_{i,s} M_{is} c_{qis} r_{qi} (1-\delta_i(k));$$

$$\forall i, \qquad \delta_i(k+1) = \left[\delta_i(k) + \epsilon \left(N \sum_q \nu_q \sum_s [\hat{M}^*(q,k)]_{is} \cdot c_{qis} r_{qi} - b_i \right) \right]^+,$$

where $\epsilon > 0$ is a fixed step-size parameter, and $[x]^+ = x$ if $x \ge 0$ or $[x]^+ = 0$ otherwise. Furthermore, defining $\hat{Q}_i(k) \triangleq \delta_i(k)/\epsilon$, the above iterative algorithm becomes

$$\forall q, \quad \hat{M}^*(q,k) \in \arg\max_{M\in\mathcal{M}_q} \sum_{i,s} M_{is} c_{qis} r_{qi} \left(\frac{1}{\epsilon} - \hat{Q}_i(k) \right);$$

$$\forall i, \quad \hat{Q}_i(k+1) = \left[\hat{Q}_i(k) + \hat{\lambda}_i(k) - b_i \right]^+,$$

where

$$\hat{\lambda}_i(k) \triangleq N \sum_q \nu_q \sum_s [\hat{M}^*(q,k)]_{is} c_{qis} r_{qi}. \quad (5)$$

Note that $\hat{Q}_i(k)$ can be interpreted as a queue which has $\hat{\lambda}_i(k)$ arrivals and b_i departures at step k. Although this algorithm already uses the feedback provided by $\{\hat{\mathbf{Q}}(k)\}$ (or $\{\boldsymbol{\delta}(k)\}$) about the state of the system, it is still using a priori information about the arrival rates of queries in $\{\hat{\boldsymbol{\lambda}}(k)\}$, hence not really "online." However, it motivates us to incorporate a queueing system with stochastic arrivals into the real online algorithm, which will be described in the next subsection.

B. Stochastic Model, Online Algorithm, and "Overdraft Queue"

In practice, a search service provider may not have a priori information about the query arrival rates $\{\nu_q\}$, and generally, query arrivals during each time slot are stochastic rather than constant. Let time slots be indexed by $t \in \mathcal{Z}^+ \cup \{0\}$. We specify our detailed statistical assumptions as follows:

- Query arrivals: Assume that a time slot is short enough so that query arrivals in each time slot can be modeled as a Bernoulli random variable with occurrence probability ν. The probability that an arrived query is for keyword q is assumed to be ϑ_q and $\sum_q \vartheta_q = 1$. Let $\tilde{q}(t)$ represent the index of the keyword queried in time slot t, such that $\tilde{q}(t) = q$ w.p. $\nu_q = \nu\vartheta_q$ for all q (indexed by positive integers) and $\tilde{q}(t) = 0$ w.p. $1-\nu$, which accounts for the case that no query arrives.
- Budget spending: We limit the values of budget spent in each budgeting cycle to be integers. To match the average budget b_i (when it is not an integer), the budget of client i in budgeting cycle k is assumed to be a random variable $\tilde{b}(k)$ which equals $\lceil b_i \rceil$ w.p. ϱ_i and $\lfloor b_i \rfloor$ otherwise, such that $E[\tilde{b}(k)] = \varrho_i \lceil b_i \rceil + (1-\varrho_i)\lfloor b_i \rfloor = b_i$, i.e., $\varrho_i = \frac{b_i - \lfloor b_i \rfloor}{\lceil b_i \rceil - \lfloor b_i \rfloor} = b_i - \lfloor b_i \rfloor$. For the trivial case that b_i is already an integer, we let $\varrho_i = 1$.
- Click-through behaviors: In time slot t, after a query for keyword q arrives, if the ad of client i is posted on webpage slot s in response to this query, then whether this ad will be clicked is modeled as a Bernoulli random variable $\tilde{c}_{qis}(t)$ with occurrence probability c_{qis}.

We now want to implement the above iterative algorithm online based on this stochastic model. According to definition (5), $\hat{\lambda}_i$ includes average query arrivals and click-through choices within N time slots (i.e., one budgeting cycle). Thus, each iteration step in the online algorithm should correspond to a budgeting cycle. For convenience, we define

$$\mathbf{u}(k) \triangleq \{\tilde{q}(t), \tilde{\mathbf{c}}(t) \text{ for } kN \le t \le kN+N-1\}$$

as a collection of random variables describing user behaviors (including stochastic query arrivals and click-through choices) in budgeting cycle k. The online algorithm is then described as follows:

Online Algorithm: (starting from $k=0$)

In each time slot $t \in [kN, kN+N-1]$, if $\tilde{q}(t) > 0$, choose the assignment matrix

$$\begin{aligned} &\tilde{M}^*(t, \tilde{q}(t), \mathbf{Q}(k)) \\ \in \quad &\arg\max_{M \in \mathcal{M}_{\tilde{q}(t)}} \sum_{i,s} M_{is} c_{\tilde{q}(t)is} r_{\tilde{q}(t)i} \left(\frac{1}{\epsilon} - Q_i(k)\right). (6)\end{aligned}$$

At the end of each budgeting cycle k, for each client i, update

$$Q_i(k+1) = \left[Q_i(k) + A_i(k, \mathbf{Q}(k), \mathbf{u}(k)) - \tilde{b}_i(k)\right]^+, \quad (7)$$

where

$$\begin{aligned} &A_i(k, \mathbf{Q}(k), \mathbf{u}(k)) \\ \triangleq &\sum_{t=kN}^{kN+N-1} \sum_s [\tilde{M}^*(t, \tilde{q}(t), \mathbf{Q}(k))]_{is} \cdot \tilde{c}_{\tilde{q}(t)is}(t) \cdot r_{\tilde{q}(t)i}. (8)\end{aligned}$$

Here, $A_i(k, \mathbf{Q}(k), \mathbf{u}(k))$ represents the revenue obtained by the service provider from client i during budgeting cycle k, and recall that $\tilde{b}_i(k)$ is a random variable which takes integer values whose mean is equal to the average budget per budgeting cycle.

In this algorithm, client i is associated with a virtual queue Q_i (maintained at the search service provider). During budgeting cycle k, the amount of money client i is charged by the search service provider $A_i(k, \mathbf{Q}(k), \mathbf{u}(k))$ is the arrival to this queue, and the average budget per budgeting cycle b_i is the departure from this queue. Note that if this queue is positive, it means that the total value of the real service already provided to the client has temporarily exceeded the client's budget, i.e., "free" service has been provided temporarily. Hence, we call this queue the "overdraft queue."

There are two different time scales here. The faster one is a time slot, the smallest time unit used to capture user behaviors (including stochastic query arrivals and click-through choices) and execute ad-posting strategies. The slower one is a budgeting cycle (equal to N time slots), at the end of which the overdraft queues are updated based on the revenue obtained over the whole budgeting cycle.

We make the following assumptions on the above stochastic model: $\{\tilde{q}(t)\}$ are i.i.d. across time slots t; $\{\tilde{c}_{qis}(t)\}$ are independent across q, i, s, and t; each variable in $\{\tilde{q}(t)\}$ and each variable in $\{\tilde{c}_{qis}(t)\}$ are mutually independent. In fact, the model can be generalized to allow for query arrivals correlated over time and across keywords, and other similar correlations inside the click-through choices or between these two stochastic processes. Such models would only make the stochastic analysis more cumbersome, but the main results will continue to hold under these more general models.

In order to guarantee that the Markov chain which we will define later is both irreducible and aperiodic, we further assume that the probability of whether there is an arrival in a time slot $\nu \in (0,1)$. We also assume that r_{qi} for all q and i can only take integer values. Together with the fact that $\tilde{b}(k)$ takes integer values, $\{\mathbf{Q}(k)\}$ becomes a discrete-time integer-valued queue. Note that assuming integer values is only for ease of analysis, but not necessary.

C. An Upper Bound on the Overdraft

According to the ad assignment step (6), if at the beginning of budgeting cycle k, $Q_i(k) > 1/\epsilon$, then for this budgeting cycle, the i^{th} row of $\tilde{M}^*(t, q, \mathbf{Q}(k))$ is always a zero vector, i.e., the service provider will not post the ads of client i until $Q_i(k)$ falls below $1/\epsilon$. Since by assumption the number of query arrivals per time slot is upper bounded, for any budgeting cycle k, one can bound the transient length of each overdraft queue as below:

$$Q_i(k) \leq \frac{1}{\epsilon} + N \cdot \arg\max_{q,s}\{r_{qi}c_{qis}\} - \lfloor b_i \rfloor, \ \forall i.$$

Therefore, $Q_i(k) \sim O(1/\epsilon)$ for all i, and stability is not an issue for these "upper bounded" queues. It further implies that this online algorithm satisfies the budget constraints in the long run, i.e.,

$$\lim_{K\to\infty} E\left[\frac{1}{K}\sum_{k=0}^{K-1} A_i(k, \mathbf{Q}(k), \mathbf{u}(k))\right] \leq b_i, \quad \forall i. \tag{9}$$

D. Near-Optimality of the Online Algorithm

We now show that, in the long run, the proposed online algorithm achieves a revenue that is close to the optimal revenue. For convenience, we use $\bar{R}(\mathbf{p})$ to denote the objective in (1), i.e.,

$$\bar{R}(\mathbf{p}) \triangleq \sum_q \nu_q \sum_{M\in\mathcal{M}_q} p_{qM} \sum_{i,s} M_{is}c_{qis}r_{qi}, \tag{10}$$

and we will use $\mathbf{p}^*$ and $\bar{R}(\mathbf{p}^*)$ in the following text to respectively represent the offline optimal solution and the offline optimal objective value for the optimization problem (1). At the same time, we define the revenue obtained during budgeting cycle k as

$$R(k) \triangleq \sum_i A_i(k, \mathbf{Q}(k), \mathbf{u}(k)).$$

To prove the convergence of the long-term average revenue to $\bar{R}(\mathbf{p}^*)$, we start with the following lemma:

Lemma 1: Consider the Lyapunov function $V(\mathbf{Q}) = \frac{1}{2}\sum_i Q_i^2$. For any $\epsilon > 0$, and each time period k,

$$\begin{aligned} & E[V(\mathbf{Q}(k+1))|\mathbf{Q}(k) = \mathbf{Q}] - V(\mathbf{Q}) \\ \leq \ & -\frac{N}{\epsilon}\left(\bar{R}(\mathbf{p}^*) - \bar{R}(\tilde{\mathbf{p}}^*(k, \mathbf{Q}))\right) + B_1 - B_2\sum_i Q_i, \end{aligned}$$

where

$$\begin{aligned} B_1 \ \triangleq \ & \frac{1}{2}\sum_i \Big((N \cdot \arg\max_{q,s}\{c_{qis}r_{qi}\})^2 + \lceil b_i \rceil^2(b_i - \lfloor b_i \rfloor) \\ & + \lfloor b_i \rfloor^2(1 - b_i + \lfloor b_i \rfloor)\Big), \end{aligned} \tag{11}$$

$$B_2 \ \triangleq \ \min_i\{b_i - N\sum_q \nu_q \sum_{M\in\mathcal{M}_q} p^*_{qM} \sum_s M_{is}c_{qis}r_{qi}\}, \tag{12}$$

and $\tilde{\mathbf{p}}^*(k, \mathbf{Q}) \triangleq \{\tilde{p}^*_{qM}(k, \mathbf{Q}), \ \forall q, M \in \mathcal{M}_q\}$ where $\tilde{p}^*_{qM}(k, \mathbf{Q})$ equals 1 if $M = \tilde{M}^*(t, q, \mathbf{Q})$ for $kN \leq t \leq kN+N-1$ (i.e., the optimal matrix in the maximization step (6)) and 0 otherwise. and $\mathbf{p}^*$ is the offline optimal solution to optimization problem (1). ◇

The proof is given in our technical report [16].

Now we are ready to present one of the major theorems in this paper, indicating that the long-term average revenue achieved by our online algorithm is within $O(\epsilon)$ of the maximum revenue obtained by the offline optimal solution. The proof is also given in our technical report [16].

Theorem 1: For any $\epsilon > 0$,

$$0 \leq \lim_{K\to\infty} E\left[\bar{R}(\mathbf{p}^*) - \frac{1}{KN}\sum_{k=0}^{K-1} R(k)\right] \leq \frac{B_1\epsilon}{N},$$

for some constant $B_1 > 0$ (defined in (11) in Lemma 1). ◇

IV. Extensions

A. Underdraft: Staying under the Budget

In the previous section, we allowed the provision of temporary free service to clients, which we call overdraft. If this is not desirable for some reason, the algorithm can be modified to have non-positive overdraft under the reasonable assumption that the number of arrivals in each time slot is bounded. We do this by allowing the queue lengths to become negative, but not positive. The practical meaning of negative queue lengths is to allow each client to accumulate a certain volume of "credits" if the current budget is under-utilized and use these credits to offset future possible overdrafts. We call this negative queue length *"underdraft."* Corresponding to this mechanism, we modify our online algorithm as follows: in response to each query for keyword q, which arrives in time slot $t \in [kN, kN + N - 1]$, choose the assignment matrix

$$\begin{aligned} & \tilde{M}^*(t, \tilde{q}(t), \mathbf{Q}(k)) \\ \in \ & \arg\max_{M\in\mathcal{M}_{\tilde{q}(t)}} \sum_{i,s} M_{is}c_{\tilde{q}(t)is}r_{\tilde{q}(t)i}\left(\Gamma_i - Q_i(k)\right), \end{aligned}$$

and at the end of budgeting cycle k, for each client i, update

$$Q_i(k+1) = \max\{Q_i(k) + A_i(k, \mathbf{Q}(k), \mathbf{u}(k)) - \tilde{b}_i(k), -C_i\},$$

where Γ_i denotes a customized "throttling threshold" (not necessarily $1/\epsilon$) and C_i denotes the maximum allowable credit volume for client i. Setting

$$\Gamma_i := \left[\lfloor b_i \rfloor - N \cdot \arg\max_{q,s}\{r_{qi}c_{qis}\}\right]^- \tag{13}$$

and $C_i := 1/\epsilon - \Gamma_i$ for all i, we can always eliminate overdrafts (i.e., $Q_i(k) \leq 0$ for all k). We can show the same near-optimal revenue by simply shifting $Q_i(k)$ to be nonnegative, i.e., $\tilde{Q}_i(k) := Q_i(k) + C_i$ for all i. Detailed discussions about the underdraft mechanism, including possible unfairness problems, can be found in our technical report [16].

B. Impact of Click-Through Rate Estimation

In our online algorithm, the decision of picking an optimal ad assignment matrix in (6) in response to each query is based on the true click-through rates $\mathbf{c}$. In reality, an estimate $\hat{\mathbf{c}}$ based on historical click-through behaviors is used, i.e., in response to each query for keyword q, which arrives in time slot $t \in [kN, kN+N-1]$, we choose the assignment matrix

$$\tilde{M}^*(t, \tilde{q}(t), \mathbf{Q}(k)) \in \arg\max_{M \in \mathcal{M}_{\tilde{q}(t)}} \sum_{i,s} M_{is} \hat{c}_{\tilde{q}(t)is} r_{\tilde{q}(t)i} \left(\frac{1}{\epsilon} - Q_i(k) \right).$$

We then have the following corollary in addition to Theorem 1 in Subsection III-D:

Corollary 1: Assume that the estimated click-through rates $\hat{\mathbf{c}} \in [\mathbf{c}(1-\Delta), \mathbf{c}(1+\Delta)]$ with some $\Delta \in (0,1)$. Under our online algorithm with estimated click-through rates, $\mathbf{Q}(k)$ is still positive recurrent. For any $\epsilon > 0$,

$$\lim_{K\to\infty} E\left[\frac{1}{KN} \sum_{k=0}^{K-1} R(k) \right] \geq \left(\frac{1-\Delta}{1+\Delta} \right) \cdot \bar{R}(\mathbf{p}^*) - \frac{B_1 \epsilon}{N},$$

for some constant $B_1 > 0$ (defined in equation (11) in Lemma 1). ◇

The proof is given in our technical report [16].

Remark 1: Corollary 1 tells us that for small ϵ, the long-term average revenue achieved by our online algorithm with estimated click-through rates will be at least $\left(\frac{1-\Delta}{1+\Delta}\right)$ of the offline optimal revenue. ◇

It is easy to see that the upper bound on the overdraft level still holds in this case.

V. Conclusions

In this paper, we propose a stochastic model to describe how search service providers charge client companies based on users' queries for the keywords related to these companies' ads by using certain advertisement assignment strategies. We formulate an optimization problem to maximize the long-term average revenue for the service provider under each client's long-term average budget constraint, and design an online algorithm which captures the stochastic properties of users' queries and click-through behaviors. We solve the optimization problem by making connections to scheduling problems in wireless networks, queueing theory and stochastic networks.

With a small customizable parameter ϵ which is the step size used in each iteration of the online algorithm, we have shown that our online algorithm achieves a long-term average revenue which is within $O(\epsilon)$ of the optimal revenue and the overdraft level of this algorithm is upper bounded by $O(1/\epsilon)$. By allowing negative values for the length of overdraft queues, we can eliminate overdraft.

When estimated click-through rates instead of true ones are used in our online algorithm, we show that the achievable fraction of the offline optimal revenue is lower bounded by $\frac{1-\Delta}{1+\Delta}$, where Δ is the relative error in click-through rate estimation.

In the work following this paper (see [16]), we further prove that in the long run, an expected overdraft level of $\Omega(\log(1/\epsilon))$ is unavoidable (a universal lower bound) under any stationary ad assignment algorithm which achieves a long-term average revenue within $O(\epsilon)$ of the offline optimum, and also show the tightness of this lower bound. Additionally, in [16] we extend our results to a click-through rate maximization model, and show how our algorithm can be modified to handle non-stationary query arrival processes and clients with short-term contracts.

References

[1] V. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press, 2008.

[2] N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. *In Proc. of the 15th annual European Conference on Algorithms (ESA), Lecture Notes in Computer Science (LNCS)*, 4698:253–264, 2007.

[3] N. R. Devenur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proc. of the 10th ACM Conference on Electronic Commerce (EC)*, pages 71–78, Stanford, CA, USA, 2009.

[4] J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. *Algorithms–ESA, Lecture Notes in Computer Science*, 6346:182–194, 2010.

[5] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. *In the 5th Workshop on Internet and Network Economics (WINE), Lecture Notes in Computer Science (LNCS)*, 5929:374–385, Dec. 2009.

[6] J. Feldman and S. Muthukrishnan. Algorithmic methods for sponsored search auctions. *SIGMETRICS Tutorial. Chapter in Performance Modeling and Engineering*, 2008.

[7] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1):1–144, 2006.

[8] G. Iyengar and A. Kumar. Characterizing optimal adword auctions. *Technical Report*, Nov. 2006. http://arxiv.org/abs/cs.GT/0611063.

[9] J. J. Jaramillo and R. Srikant. Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic. In *Proc. of IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010.

[10] B. Kalyanasundaram and K. Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1–2):319–325, 2000.

[11] R. Karp, U. Vazirani, and V. Vazirani. An optimal algorithm for online bipartite matching. In *Proc. of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 352–358, 1990.

[12] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, Aug. 2006.

[13] M. Mahdian, H. Nazerzadeh, and A. Saberi. Allocating online advertisement space with unreliable estimates. In *Proc. of the 8th ACM conference on Electronic Commerce (EC)*, pages 288–294, 2007.

[14] A. Mehta, A. Saberi, U. V. Vazirani, and V. Vazirani. Adwords and generalized online matching. In *Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 264–273, 2005.

[15] I. Menache, A. Ozdaglar, R. Srikant, and D. Acemoglu. Dynamic online-advertising auctions as stochastic scheduling. In *Proc. of the Workshop on the Economics of Networks, Systems and Computation (NetEcon)*, Stanford, CA, USA, Jul. 2009.

[16] B. R. Tan and R. Srikant. Online advertisement, optimization and stochastic networks. *Technical Report*, June 2011. (http://arxiv.org/abs/1009.0870).