Online Competitive Algorithms for Ad Allocation in Cellular Networks (AdCell) ⋆

Saeed Alaei§[1], Mohammad T. Hajiaghayi¶[12], Vahid Liaghat‖[1],
Dan Pei[2], and Barna Saha**[1]
{saeed, hajiagha, vliaghat, barna}@cs.umd.edu, peidan@research.att.com

[1] University of Maryland, College Park, MD, 20742
[2] AT&T Labs - Research, 180 Park Avenue, Florham Park, NJ 07932

Abstract. With more than four billion usage of cellular phones worldwide, mobile advertising has become an attractive alternative to online advertisements. In this paper, we propose a new targeted advertising policy for Wireless Service Providers (WSPs) via SMS or MMS- namely *AdCell*. In our model, a WSP charges the advertisers for showing their ads. Each advertiser has a valuation for specific types of customers in various times and locations and has a limit on the maximum available budget. Each query is in the form of time and location and is associated with one individual customer. In order to achieve a non-intrusive delivery, only a limited number of ads can be sent to each customer. Recently, new services have been introduced that offer location-based advertising over cellular network that fit in our model (e.g., ShopAlerts by AT&T) .

We consider both online and offline version of the AdCell problem and develop approximation algorithms with constant competitive ratio. For the online version, we assume that the appearances of the queries follow a stochastic distribution and thus consider a Bayesian setting. Furthermore, queries are not i.i.d. and they may come from different distributions on different times. This model generalizes several previous advertising models such as online secretary problem [8], online bipartite matching [12,6] and AdWords [19]. Since our problem generalizes the well-known secretary problem, no non-trivial approximation can be guaranteed in the online setting without stochastic assumptions. We propose an online algorithm that is simple, intuitive and easily implementable in practice. It is based on pre-computing a fractional solution for the expected scenario and relies on a novel use of dynamic programming to compute the conditional expectations. We show that our algorithms are $\frac{1}{2}$ and $1 - \frac{1}{e}$-competitive when we have constraints on capacities and budgets respectively. We give tight lower bounds on the approximability of some variants of the problem as well. In the offline setting we achieve near-optimal bounds, matching the integrality gap of the considered linear program. We believe that our proposed solutions can be used for other advertising settings where personalized advertisement is critical.

Keywords: Mobile Advertisement, AdCell, Online, Matching

⋆ A preliminary version of this paper appeared in European Symposium on Algorithms, Saarbrücken, Germany, Sep 2011.
§ Supported in part by NSF Grant CCF-0728839.
¶ Supported in part by NSF CAREER Award and ONR Young Investigator Award.
‖ Supported in part by NSF CAREER Award and Google Faculty Research Award.
** Supported in part by NSF Award CCF-0728839, NSF Award CCF-0937865.

1 Introduction

In this paper, we propose a new mobile advertising concept called *Adcell*. More than 4 billion cellular phones are in use world-wide, and with the increasing popularity of smart phones, mobile advertising holds the prospect of significant growth in the near future. Some research firms [1] estimate mobile advertisements to reach a business worth over 10 billion US dollars by 2012. Given the built-in advertisement solutions from popular smart phone OSes, such as iAds for Apple's iOS, mobile advertising market is poised with even faster growth.

In the mobile advertising ecosystem, wireless service providers (WSPs) render the physical delivery infrastructure, but so far WSPs have been more or less left out from profiting via mobile advertising because of several challenges. First, unlike web, search, application, and game providers, WSPs typically do not have users' application context, which makes it difficult to provide targeted advertisements. Deep Packet Inspection (DPI) techniques that examine packet traces in order to understand application context, is often not an option because of privacy and legislation issues (i.e., Federal Wiretap Act). Therefore, a targeted advertising solution for WSPs need to utilize *only the information it is allowed to collect by government and by customers via opt-in mechanisms*. Second, without the luxury of application context, targeted ads from WSPs require *non-intrusive delivery methods*. While users are familiar with other ad forms such as banner, search, in-application, and in-game, push ads with no application context (e.g., via SMS) can be intrusive and annoying if not done carefully. The number and frequency of ads both need to be well-controlled. Third, targeted ads from WSPs should be well personalized such that the users have incentive to read the advertisements and take purchasing actions, especially given the requirement that the number of ads that can be shown to a customer is limited.

In this paper, we propose a new mobile targeted advertising strategy, *AdCell*, for WSPs that deals with the above challenges. It takes advantage of the detailed real-time location information of users. Location can be tracked upon users' consent. This is already being done in some services offered by WSPs, such as Sprint's Family Location and AT&T's Family Map, thus there is no associated privacy or legal complications. To locate a cellular phone, it must emit a roaming signal to contact some nearby antenna tower, but the process does not require an active call. GSM localization is then done by multi-lateration[3] based on the signal strength to nearby antenna masts [21]. Location-based advertisement is not completely new. Foursquare mobile application allows users to explicitly "check in" at places such as bars and restaurants, and the shops can advertise accordingly. Similarly there are also automatic proximity-based advertisements using GPS or bluetooth. For example, some GPS models from Garmin display ads for the nearby business based on the GPS locations [22]. ShopAlerts by AT&T [4] is another application along the same line. On the advertiser side, popular stores such as Starbucks are reported to have attracted significant footfalls via mobile coupons.

Most of the existing mobile advertising models are On-Demand, however, AdCell sends the ads via SMS, MMS, or similar methods without any prior notice. Thus to deal with the non-intrusive delivery challenge, we propose user subscription to advertising services that deliver only a *fixed number* of ads per month to its subscribers (as it is the case in AT&T ShopAlerts). The constraint of delivering limited number of ads to each customer adds the main algorithmic challenge in the AdCell model (details in Section 1.1). In order to overcome the incentive challenge, the WSP can "pay" users to read ads and purchase based on them through a reward program in the form of credit for monthly wireless bill. To begin with, both customers and advertisers should sign-up for the AdCell-service provided by the WSP (e.g., currently there are 9 chain-companies participating in ShopAlerts). Customers enrolled for the service should sign an agreement that their *location* information will be tracked; but solely for the advertisement purpose. Advertisers (e.g., stores) provide their advertisements and a maximum chargeable budget to the WSP. The WSP selects proper ads (these, for example, may depend on time and distance of a customer from a store) and sends them (via SMS) to the customers. The WSP charges the advertisers for showing their ads and also for successful ads. An ad is deemed successful if a customer visits the advertised store. Depending on the service plan, customers are entitled to receive different number of advertisements per month. Several logistics need to be employed to improve AdCell experience and enthuse customers into participation. We provide more details about these logistics in Appendix A.

1.1 AdCell Model & Problem Formulation

In the AdCell model, advertisers bid for individual customers based on their location and time. The triple (k, ℓ, t) where k is a customer, ℓ is a neighborhood (location) and t is a time forms a *query* and there is a bid amount (possibly zero) associated with each query for each advertiser. This definition of query allows advertisers to customize their bids based on customers, neighborhoods and time. We assume a customer can only be in one neighborhood

[3] The process of locating an object by accurately computing the time difference of arrival of a signal emitted from that object to three or more receivers.

[4] http://shopalerts.att.com/sho/att/index.html

at any particular time and thus at any time t and for each customer k, the queries (k, ℓ_1, t) and (k, ℓ_2, t) are mutually exclusive, for all distinct l_1, l_2. Neighborhoods are places of interest such as shopping malls, airports, etc. We assume that queries are generated at certain times (e.g., every half hour) and only if a customer stays within a neighborhood for a specified minimum amount of time. The formal problem definition of *AdCell Allocation* is as follows:

AdCell Allocation *There are m advertisers, n queries and s customers. Advertiser i has a total budget b_i and bids u_{ij} for each query j. Furthermore, for each customer $k \in [s]$, let S_k denote the queries corresponding to customer k and c_k denote the maximum number of ads which can be sent to customer k. The capacity c_k is associated with customer k and is dictated by the AdCell plan the customer has signed up for. Advertiser i pays u_{ij} if his advertisement is shown for query j and if his budget is not exceeded. That is, if x_{ij} is an indicator variable set to 1, when advertisement for advertiser i is shown on query j, then advertiser i pays a total amount of $\min(\sum_j x_{ij}u_{ij}, b_i)$. The goal of AdCell Allocation is to specify an advertisement allocation plan such that the total payment $\sum_i \min(\sum_j x_{ij}u_{ij}, b_i)$ is maximized.*

The AdCell problem is a generalization of the budgeted AdWords allocation problem [4,20] with capacity constraint on each customer and thus is NP-hard. In this paper, we mainly concentrate on the online version where queries arrive online and a decision to assign a query to an advertiser has to be done right away. With arbitrary queries/bids and optimizing for the worst case, one cannot obtain any approximation algorithm with ratio better than $\frac{1}{n}$. This follows from the observation that online AdCell problem also generalizes the *secretary problem* for which no deterministic or randomized online algorithm can get approximation ratio better than $\frac{1}{n}$ in the worst case.[5]. Therefore, we consider a stochastic setting.

For the online AdCell problem, we assume that each query j arrives with probability p_j. Upon arrival, each query has to be either allocated or discarded right away. We note that each query encodes a customer id, a location id and a time stamp. Also associated with each query, there is a probability, and a vector consisting of the bids for all advertisers for that query. Furthermore, we assume that all queries with different arrival times or from different customers are independent, however queries from the same customer with the same arrival time are mutually exclusive (i.e., a customer cannot be in multiple locations at the same time). Note that if a customer is at a specific location, it is more likely that in the next period of time he is at a neighboring location. We currently do not consider such correlated queries, and leave this as a future work.

There is a crucial difference between the AdCell model and the previous models studied in the literature for AdWords allocation. In the adword allocation problem the arrival of adwords (queries) is i.i.d., e.g., we assume we may see the adword "lunch" in the morning with the same probability that we may see it at noon. However, in AdCell the queries are in the form of the triple (customer, location, and *time*) and thus a query (k, ℓ, t) can only be revealed at time t. This is a more general and realistic assumption since for example for a location "school" we may have different probability distributions for different times, e.g., for a student k the probability of the query $(k, \textit{school}, \textit{midnight})$ is zero but the probability of the query $(k, \textit{school}, \textit{noon})$ is high, hence the bidders can also customize their bid values based on time. This shows that AdCell is a generalization of *Prophet Inequality* over bipartite matchings.

The topic of Prophet Inequality has been studied in optimal stopping theory since the 1970s [14,15,13] and more recently in computer science [9]. In prophet inequality setting, given the distribution of a sequence of random variables $x_1, \cdots, x_n$ an onlooker has to choose from the succession of these values, where x_t is revealed to us at time t. The onlooker can only choose a certain number of values (called her capacity) and cannot choose a past value. The onlooker's goal is to maximize her revenue. The inequality has been interpreted as meaning that a prophet with complete foresight has only a bounded advantage over an onlooker who observes the random variables one by one, and this explains the name Prophet Inequality. We note that the prophet inequality problem is a special case of our AdCell model in which all the budgets are infinity and there is only one customer. The capacity of the customer would correspond to the number of values the onlooker can choose in the prophet inequality problem.

1.2 Our Results and Techniques

Here we provide a summary of our results and techniques. We consider primarily the online version of the problem. We only know the arrival probabilities of queries (i.e., $p_1, \cdots, p_m$).

[5] The reduction of the *secretary problem* to AdCell problem is as follows: consider a single advertiser with large enough budget and a single customer with a capacity of 1. The queries correspond to secretaries and the bids correspond to the values of the secretaries. So we can only allocate one query to the advertiser.

We can write the AdCell problem as the following random integer program in which $\mathbf{I}_j$ is the indicator random variable which is 1 if query j arrives and 0 otherwise:

$$\begin{aligned}
&\text{maximize.} && \sum_i \min(\sum_j \mathbf{X}_{ij} u_{ij}, b_i) && (IP_{BC})\\
&\forall j \in [n]: && \sum_i \mathbf{X}_{ij} \le \mathbf{I}_j && (F)\\
&\forall k \in [s]: && \sum_{j \in S_k} \sum_i \mathbf{X}_{ij} \le c_k && (C)\\
& && \mathbf{X}_{ij} \in \{0,1\}
\end{aligned}$$

In the above integer program, X_{ij} is set to 1 if query j arrives and advertiser i is allocated query j. (C) denote the capacity constraints. We refer to the variant of the problem explained above as IP_{BC}. We also consider variants in which there are either budget constraints or capacity constraints but not both. We refer to these variants as IP_B and IP_C respectively. The above integer program can be relaxed to obtain a linear program LP_{BC}, where we maximize $\sum_i \sum_j \mathbf{X}_{ij} u_{ij}$ with the constraints (F), (C) and additional budget constraint $\sum_j \mathbf{X}_{ij} u_{ij} \le b_i$ which we refer to by (B). We relax $\mathbf{X}_{ij} \in \{0,1\}$ to $\mathbf{X}_{ij} \in [0,1]$.

We also refer to the variant of this linear program with only either constraints of type (B) or constraints of type (C) as LP_B and LP_C.

In the online version, we assume to know the $E[\mathbf{I}_j]$ in advance and we learn the actual value of $\mathbf{I}_j$ online. We note a crucial difference between our model and the i.i.d model. In i.i.d model the probability of the arrival of a query is independent of the time, i.e., queries arrive from the same distribution on each time. However, in AdCell model a query encodes time (in addition to location and customer id), hence we may have a different distribution on each time. This implies a prophet inequality setting in which on each time, an onlooker has to decide according to a given value where this value may come from a different distribution on different times (e.g. see [14,9]).

In the online version, we compare the expected revenue of our solution with the expected revenue of the optimal offline algorithm. We obtain the following results.

- We provide a $\frac{1}{2}\left(1-\frac{1}{e}\right)$-approximation algorithm for the online version with both capacity and budget constraints, that is for the AdCell problem (Section 3).
- We provide a $\left(1-\frac{1}{e}\right)$-approximation algorithm when there is only budget constraints (Section 3).
- We provide a $\frac{1}{2}$-approximation algorithm when there is only capacity constraints. (Section 3).

We should emphasis that we make no assumptions about bid to budget ratios (e.g., bids could be as large as budgets).

To draw a comparison to the results known for the Prophet Inequality, note that the famous Prophet Inequality by Krengel, Sucheston, and Garling in 1970's concerns the case in which the values are chosen independently from known distributions (but not necessary identical) and the onlooker can only choose one value. Using a very simple example, they showed no online algorithm can be better than $\frac{1}{2}$-competitive [14]. Therefore one may consider our $\frac{1}{2}$-approximation algorithm when only capacity constraints are present as a generalization of the Prophet Inequality settings of Krengel, Sucheston, and Garling. We have a more complex setting where we have many customers, and additionally, budget constraints over the advertisers.

We now briefly describe our main techniques.

Breaking into smaller sub-problems that can be optimally solved using conditional expectation. Theoretically, ignoring the computational issues, any online stochastic optimization problem can be solved optimally using conditional expectation as follows: At any time a decision needs to be made, compute the total expected objective conditioned on each possible decision, then chose the one with the highest total expectation. These conditional expectations can be computed by backward induction, possibly using a dynamic program. However for most problems, including the AdCell problem, the size of this dynamic program is exponential which makes it impractical. We avoid this issue by using a randomized strategy to break the problem into smaller subproblems such that each subproblem can be solved by a quadratic dynamic program.

Using an LP to analyze the performance of an optimal online algorithm against an optimal offline fractional solution. Note that we compare the expected objective value of our algorithm against the expected objective value of the optimal offline fractional solution. Therefore for each subproblem, even though we use an optimal online algorithm, we still need to compare its expected objective value against the expected objective value of the optimal offline solution for that subproblem. Basically, we need to compare the expected objective of an stochastic online algorithm, which works by maximizing conditional expectation at each step, against the expected objective value of its optimal offline solution. To do this, we create a minimization linear program that encodes the dynamic program and whose optimal objective is the minimum ratio of the expected objective value of the online algorithm to the expected objective value of the optimal offline solution. We then prove a lower bound of $\frac{1}{2}$ on the objective value of this linear program by constructing a feasible solution for its dual obtaining an objective value of $\frac{1}{2}$.

2 Related Work

Online advertising alongside search results is a multi-billion dollar business [16] and is a major source of revenue for search engines like Google, Yahoo and Bing. A related ad allocation problem is the AdWords assignment problem [19] that was motivated by sponsored search auctions. When modeled as an online bipartite assignment problem, each edge has a weight, and there is a budget on each advertiser representing the upper bound on the total weight of edges that might be assigned to it. In the offline setting, this problem is NP-Hard, and several approximations have been proposed [3,2,4,20]. For the online setting, it is typical to assume that edge weights (i.e., bids) are much smaller than the budgets, in which case there exists a $(1 - 1/e)$-competitive online algorithm [19]. Recently, Devanur and Hayes [5] improved the competitive ratio to $(1 - \epsilon)$ in the stochastic case where the sequence of arrivals is a random permutation.

Another related problem is the online bipartite matching problem which is introduced by Karp, Vazirani, and Vazirani [12]. They proved that a simple randomized online algorithm achieves a $(1 - 1/e)$-competitive ratio and this factor is the best possible. Online bipartite matching has been considered under stochastic assumptions in [7,6,18], where improvements over $(1 - 1/e)$ approximation factor have been shown. The most recent of of them is the work of Manshadi et al. [18] that presents an online algorithm with a competitive ratio of 0.702. They also show that no online algorithm can achieve a competitive ratio better than 0.823. More recently, Mahdian et al.[17] and Mehta et al.[11] improved the competitive ratio to 0.696 for unknown distributions.

3 Online Setting

In this section, we present three online algorithms for the three variants of the problem mentioned in the pervious section (i.e., IP_B, IP_C and IP_{BC}).

First, we present the following lemma which provides a means of computing an upper bound on the expected revenue of any algorithm for the AdCell problem.

Lemma 1 (Expectation Linear Program). *Consider a general random linear program in which* $\mathfrak{b}$ *is a vector of random variables:*

(Random LP)

maximize. $\quad c^T x$

s.t. $\quad Ax \leq \mathfrak{b}; \;\; x \geq 0$

Let $OPT(\mathfrak{b})$ *denote the optimal value of this program as a function of the random variables. Now consider the following linear program:*

(Expectation LP)

maximize. $\quad c^T x$

s.t. $\quad Ax \leq E[\mathfrak{b}]; \;\; x \geq 0$

We refer to this as the "Expectation Linear Program" *corresponding to the* "Random Linear Program". *Let* $\overline{OPT}$ *denote the optimal value of this program. Assuming that the original linear program is feasible for all possible draws of the random variables, it always holds that* $E[OPT(\mathfrak{b})] \leq \overline{OPT}$.

Proof. Let $x^*(\mathfrak{b})$ denote the optimal assignment as a function of $\mathfrak{b}$. Since the random LP is feasible for all realizations of $\mathfrak{b}$, we have $Ax^*(\mathfrak{b}) \leq \mathfrak{b}$. Taking the expectation from both sides, we get $AE[x^*(\mathfrak{b})] \leq E[\mathfrak{b}]$. So, by setting $x = E[x^*(\mathfrak{b})]$ we get a feasible solution for the expectation LP. Furthermore, the objective value resulting from this assignment is equal to the expected optimal value of the random LP. The optimal value of the expectation LP might however be higher so its optimal value is an upper bound on the expected optimal value of random LP. □

As we will see next, not only does the expectation LP provide an upper bound on the expected revenue of the optimal offline allocation, it also leads to a good approximate algorithm for the online allocation. We adopt the notation of using an overline to denote the expectation linear program corresponding to a random linear program (e.g. $\overline{LP}_{BC}$ for LP_{BC}). Next we present an online algorithm for the variant of the problem in which there are only budget constrains but not capacity constraints.

3.1 Online Algorithm with Only Budget Constraints

Algorithm 1 (STOCHASTIC ONLINE ALLOCATOR FOR IP_B)

- *Compute an optimal assignment for the corresponding expectation LP (i.e. $\overline{LP}_B$). Let x^*_{ij} denote this assignment. Note that x^*_{ij} might be a fractional assignment.*
- *If query j arrives, pick an advertiser $i \in [m]$ with probability $\frac{x^*_{ij}}{p_j}$, and allocate the query to it.*

Theorem 1. *The expected revenue of Algorithm 1 is at least $1 - \frac{1}{e}$ of the optimal value of the expectation LP (i.e., $\overline{LP}_B$) which implies that the expected revenue of 1 it is at least $1 - \frac{1}{e}$ of the expected revenue of the optimal offline allocation too. Note that this result holds even if u_{ij}'s are not small compared to b_i.*

Furthermore, the result of Theorem 1 holds even if we relax the independence requirement among the queries in the original problem and require negative correlation instead.

Two random variables $\mathbf{X}, \mathbf{Y}$ are called negatively correlated, if $cov(\mathbf{X}, \mathbf{Y}) := E[\mathbf{XY}]E[\mathbf{X}]E[\mathbf{Y}] \leq 0$. The following definition from [10] is a natural generalization to the case of negative correlation to the case of n random variables.

Definition 1 (Negative Correlation). *The random variables $\mathbf{X} = (\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_n})$ are negatively correlated (associated) if for every index set $I \subseteq [n]$, $cov(f(X_i, i \in I), g(X_i, i \in \bar{I})) \leq 0$, that is,*

$$E[f(X_i, i \in I), g(X_i, i \in \bar{I})] \leq E[f(X_i, i \in I)]E[g(X_i, i \in \bar{I})]$$

for all non-decreasing functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$, and $g : \mathbb{R}^{n-|I|} \to \mathbb{R}$. (A function $h : \mathbb{R}^k \to \mathbb{R}$ is said to be non-decreasing, if $h(\boldsymbol{x}) \leq h(\boldsymbol{y})$ whenever $\boldsymbol{x} \leq \boldsymbol{y}$ in the component-wise ordering of $\mathbb{R}^k$.)

Allowing negative correlation instead of independence makes the above model much more general. For example, suppose there is a customer, location tuple that may arrive at several different times but may only arrive at most once or only a limited number of times, we can model this by creating a new query for each possible time instance the customer and location pair can arrive. These queries are negatively correlated.

Remark 1. It is worth mentioning that there is an integrality gap of $1 - \frac{1}{e}$ between the optimal value of the integral allocation and the optimal value of the expectation LP. So the lower bound of Theorem 1 is tight. To see this, consider a single advertiser and n queries. Suppose $p_j = \frac{1}{n}$ and $u_{1j} = 1$ for all j. The optimal value of $\overline{LP}_B$ is 1 but even the expected optimal revenue of the offline optimal allocation is $1 - \frac{1}{e}$ when $n \to \infty$ because with probability $(1 - \frac{1}{n})^n$ no query arrives.

To prove Theorem 1, we use the following theorem:

Theorem 2. *Let C be an arbitrary positive number and let $\mathbf{X}_1, \cdots, \mathbf{X}_n$ be independent random variables (or negatively correlated) such that $\mathbf{X}_i \in [0, C]$. Let $\mu = E[\sum_i \mathbf{X}_i]$. Then:*

$$E[\min(\textstyle\sum_i \mathbf{X}_i, C)] \geq (1 - \frac{1}{e^{\mu/C}})C$$

Furthermore, if $\mu \leq C$ then the right hand side is at least $(1 - \frac{1}{e})\mu$.

Proof (Theorem 2). Define the random variables $\mathbf{R}_i = \max(\mathbf{R}_{i-1} - \mathbf{X}_i, 0)$ and $\mathbf{R}_0 = C$. Observe that for each i, $\mathbf{R}_i = \max(C - \sum_{j=1}^i \mathbf{X}_j, 0)$; so $\min(\sum_{j=1}^i \mathbf{X}_j, C) + \mathbf{R}_i = C$. Therefore $E[\min(\sum_{j=1}^i \mathbf{X}_j, C)] + E[\mathbf{R}_i] = C$ and to prove the theorem it is enough to show that $E[\mathbf{R}_n] \leq \frac{1}{e^{\mu/C}} \cdot C$. To show this we will prove the following inequality:

$$E[\mathbf{R}_i] \leq (1 - \frac{E[\mathbf{X}_i]}{C})E[\mathbf{R}_{i-1}] \tag{U}$$

Assuming that (U) is true, we can conclude the following which proves the claim.

$$\begin{aligned} E[\mathbf{R}_n] &\leq C \cdot \prod_{i=1}^n (1 - \frac{E[\mathbf{X}_i]}{C}) \\ &\leq C \cdot \prod_{i=1}^n e^{-E[X_i]/C} = Ce^{-\sum_{i=1}^n E[X_i]/C} \\ &= C \cdot \frac{1}{e^{\mu/C}} \end{aligned}$$

The second inequality follows from the fact that $1 - x \leq e^{-x}$ and the last equality follows by observing that $\sum_i \frac{E[\mathbf{X}_i]}{C} = \frac{\mu}{C}$.

Furthermore, to prove the second claim, that is if $\mu \leq C$ then $(1 - \frac{1}{e^{\mu/C}}) \geq (1 - \frac{1}{e})\mu$, we can use the fact that for $a \leq 1$, $(1 - e^{-a}) = (1 - (1 - a + a^2/2! - a^3/3! - ...) = a(1 - a/2! + a^2/3! - ...) = a((1-a) + a(1 - 1/2! + 1/3! - ...)) = a((1-a) + a(1 - \frac{1}{e})) = a(1 - \frac{a}{e}) \geq a(1 - 1/e)$. Therefore, $(1 - \frac{1}{e^{\mu/C}}) \cdot C \geq (1 - \frac{1}{e})\frac{\mu}{C}C = (1 - \frac{1}{e})\mu$ whenever $\mu \leq C$. Now it only remains to prove the inequality (U):

$$\begin{aligned}
E[\mathbf{R}_i] &= E[\max(\mathbf{R}_{i-1} - \mathbf{X}_i, 0)] \\
&\leq E[\max(\mathbf{R}_{i-1} - \mathbf{X}_i \frac{\mathbf{R}_{i-1}}{C}, 0)] \\
&= E[\mathbf{R}_{i-1} - \mathbf{X}_i \frac{\mathbf{R}_{i-1}}{C}] \\
&= E[\mathbf{R}_{i-1}] - \frac{1}{C} E[\mathbf{X}_i \mathbf{R}_{i-1}]
\end{aligned}$$

Suppose $E[\mathbf{X}_i \mathbf{R}_{i-1}] \geq E[\mathbf{X}_i] E[\mathbf{R}_{i-1}]$ then:

$$\begin{aligned}
&\leq E[\mathbf{R}_{i-1}] - \frac{1}{C} E[\mathbf{X}_i] E[\mathbf{R}_{i-1}] \\
&= (1 - \frac{E[\mathbf{X}_i]}{C}) E[\mathbf{R}_{i-1}]
\end{aligned}$$

This gives us the desired result.

Therefore, we are only left with proving $E[\mathbf{X}_i \mathbf{R}_{i-1}] \geq E[\mathbf{X}_i] E[\mathbf{R}_{i-1}]$. This clearly holds when $X_1, X_2, ..., X_n$ are independent random variables. Now consider the case when they are negatively correlated. We have $\mathbf{R}_{i-1} = \max(C - \sum_{j=1}^{i-1} \mathbf{X}_j, 0)$. Hence if the values of $\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_{i-1}}$ increases, the value of $\mathbf{R_{i-1}}$ does not increase. Therefore, $-\mathbf{R_{i-1}}$ is a non-decreasing function of $\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_{i-1}}$. Therefore, by the definition of negative correlation (Definition 1) $E[-\mathbf{R_{i-1}} \mathbf{X}_i] \leq E[-\mathbf{R_{i-1}}] E[\mathbf{X}_i]$, and hence $E[\mathbf{R_{i-1}} \mathbf{X}_i] \geq E[\mathbf{R_{i-1}}] E[\mathbf{X}_i]$. This completes the proof. □

Now we prove Theorem 1 using the above theorem.

Proof (Theorem 1). We apply Theorem 2 to each advertiser i separately. From the perspective of advertiser i, each query is allocated to her with probability x^*_{ij} and by constraint (B) we can argue that have $\mu = \sum_j x^*_{ij} u_{ij} \leq b_i = C$ so $\mu \leq C$ and by Theorem 2, the expected revenue from advertiser i is at least $(1 - \frac{1}{e})(\sum_j x^*_{ij} u_{ij})$. Therefore, overall, we achieve at least $1 - \frac{1}{e}$ of the optimal value of the expectation LP and that completes the proof.

Next we present an online algorithm for the variant of the problem in which there are only capacity constrains but not budget constraints.

3.2 Online Algorithm with Only Capacity Constraints

The main difference between Algorithm 1 and Algorithm 2 is that in the former whenever we choose an advertiser at random, we always allocate the query to that advertiser (assuming they have enough budget). However, in the latter, we run a dynamic program for each customer k and once an advertiser is picked at random, the query is allocated to this advertiser only if doing so increases the expected revenue associated with customer k. Below we give details of the algorithm, and how the dynamic program is run on each customer.

Algorithm 2 (STOCHASTIC ONLINE ALLOCATOR FOR IP_C)

- *Compute an optimal assignment for the corresponding expectation LP (i.e. $\overline{LP}_C$). Let x^*_{ij} denote this assignment. Note that x^*_{ij} might be a fractional assignment.*
- *Partition the items to sets $T_1, \cdots, T_u$ in increasing order of their arrival time and such that all of the items in the same set have the same arrival time.*
- *For each $k \in [s], t \in [u], r \in [c_k]$, let $E^r_{k,t}$ denote the expected revenue of the algorithm from queries in S_k (i.e., associated with customer k) that arrive at or after T_t and assuming that the remaining capacity of customer k is r. We formally define $E^r_{k,t}$ later.*

– *If query j arrives then choose one of the advertisers at random with advertiser i chosen with a probability of $\frac{x^*_{ij}}{p_j}$. Let k and T_t be respectively the customer and the partition which query j belongs to. Also, let r be the remaining capacity of customer k (i.e. r is c_k minus the number of queries from customer k that have been allocated so far). If $u_{ij} + E^{r-1}_{k,t+1} \geq E^r_{k,t+1}$ then allocate query j to advertiser i otherwise discard query j.*

We can now define $E^r_{k,t}$ recursively as follows:

$$E^r_{k,t} = \sum_{j \in T_t} \sum_{i \in [m]} x^*_{ij} \max(u_{ij} + E^{r-1}_{k,t+1}, E^r_{k,t+1}) + (1 - \sum_{j \in T_t} \sum_{i \in [m]} x^*_{ij}) E^r_{k,t+1} \quad \text{(EXP}_k\text{)}$$

Also define $E^0_{k,t} = 0$ and $E^r_{k,u+1} = 0$. Note that we can efficiently compute $E^r_{k,t}$ using dynamic programming.

Theorem 3. *The expected revenue of Algorithm 2 is at least $\frac{1}{2}$ of the optimal value of the expectation LP (i.e., $\overline{LP}_C$) which implies that the expected revenue of Algorithm 2 is at least $\frac{1}{2}$ of the expected revenue of the optimal offline allocation for IP_C too.*

Remark 2. The approximation ratio of 2 is tight. There is no online algorithm that can achieve in expectation better than $\frac{1}{2}$ of the revenue of the optimal offline allocation without making further assumptions. We show this by providing a simple example. Consider an advertiser with a large enough budget and a single customer with a capacity of 1 and two queries. The queries arrive independently with probabilities $p_1 = 1 - \epsilon$ and $p_2 = \epsilon$ with the first query having an earlier arrival time. The advertiser has submitted the bids $b_{11} = 1$ and $b_{12} = \frac{1-\epsilon}{\epsilon}$. Observe that no online algorithm can get a revenue better than $(1 - \epsilon) \times 1 + \epsilon^2 \frac{1-\epsilon}{\epsilon} \approx 1$ in expectation because at the time query 1 arrives, the online algorithm does not know whether or not the second query is going to arrive and the expected revenue from the second query is just $1 - \epsilon$. However, the optimal offline solution would allocate the second query if it arrives and otherwise would allocate the first query so its revenue is $\epsilon \frac{1-\epsilon}{\epsilon} + (1 - \epsilon)^2 \times 1 \approx 2$ in expectation.

Before we prove Theorem 3, we first define a simple stochastic knapsack problem which will be used as a building block for the proof.

Definition 2 (Stochastic Uniform Knapsack). *There is a knapsack of capacity C and a sequence of n possible items. Each item j is of size 1, has a value of v_j and arrives with probability p_j. Let $\mathbf{I}_j$ denote the indicator random variable indicating the arrival of item j. We assume that items can be partitioned into sets $T_1, \cdots, T_u$ based on their arrival times such that all the items in the same partition have the same arrival time and are mutually exclusive (i.e. at most one of them arrives) and items from different partitions are independent. Furthermore, we assume that $\sum_{j \in [n]} p_j \leq C$.*

The following algorithm based on conditional expectation computes the optimal online allocation for this problem:

Algorithm 3 (STOCHASTIC UNIFORM KNAPSACK - OPTIMAL ONLINE ALLOCATOR)
Consider a stochastic uniform knapsack problem as defined in Definition 2.

– *For each $t \in [u]$ and $r \in [C]$, let E^r_t denote the expected revenue of the algorithm from queries that arrive at or after time t (i.e. $T_t, \cdots, T_u$) and assuming that the remaining capacity of the knapsack is r. We formally define E^r_t later.*
– *If item j arrives do the following. Let t be the index of the partition which j belongs to and let r be the remaining capacity of the knapsack. Put item j in the knapsack if $v_j + E^{r-1}_{t+1} \geq E^r_{t+1}$.*

E^r_t can be defined recursively as follows and can be efficiently computed using dynamic programming:

$$E^r_t = \sum_{j \in T_t} p_j \max(v_j + E^{r-1}_{t+1}, E^r_{t+1}) + (1 - \sum_{j \in T_t} p_j) E^r_{t+1} \quad \text{(EXP)}$$

Also define $E^0_t = 0$ and $E^r_{u+1} = 0$.

Clearly the above algorithm achieves the best revenue that any online algorithm can achieve in expectation for the stochastic uniform knapsack. However, we need a stronger result since we need to compare its revenue against the optimal value of the expectation LP.

Lemma 2. *Consider the stochastic uniform knapsack problem as defined in Definition 2. Let O_o denote the random variable representing the expected revenue of Algorithm 3 for this problem (i.e. $O_o = E_1^C$, since C is the capacity of the knapsack). Also define $O_e = \sum_j p_j v_j$. Assuming that $\sum_j p_j \le C$, the following always holds:*

$$\frac{1}{2} O_e \le E[O_o] \le O_e$$

Proof (Theorem 3). We apply Lemma 2 to the subset of queries associated with each customer k (i.e. S_k) separately. We may think of this as having a knapsack of capacity c_k for customer k. Each pair of advertiser/query, (i, j) is a knapsack item with value u_{ij}. All knapsack items of the form (i, j) with the same j are mutually exclusive (because at most one advertiser is chosen at random) and they all have the same arrival time. Therefore, by applying Lemma 2, from the knapsack of each customer k we get at least $\frac{1}{2}(\sum_{j \in S_k} \sum_i x^*_{ij} u_{ij})$ in expectation. So overall, we get $\frac{1}{2}$ of the optimal value of the expectation LP and that completes the proof.

3.3 Online Algorithm with both Capacity and Budget Constraints

Next, we show that an algorithm similar to the one with only capacity constraints can be used when there are both budget and capacity constraints, that is for the AdCell problem.

Algorithm 4 (STOCHASTIC ONLINE ALLOCATOR FOR IP_{BC})
Run the same algorithm as in 2 except that now x^*_{ij} is a fractional solution of $\overline{LP}_{BC}$ instead of $\overline{LP}_C$.

Theorem 4. *The expected revenue of Algorithm 4 is at least $\frac{1}{2} - \frac{1}{e}$ of the optimal value of the expectation LP (i.e., $\overline{LP}_{BC}$) which implies that the expected revenue of Algorithm 4 is at least $\frac{1}{2}(1 - \frac{1}{e})$ of the expected revenue of the optimal offline allocation too.*

Proof. The proof is essentially the same as the proof of Theorem 3. The only difference is that we may also lose at most a factor of $\frac{1}{e}$ from each advertiser due to going over the budget limit.

From the point of view of advertiser i, on arrival of query j, he is chosen with probability x^*_{ij}, and is allocated query j with probability (say) y^*_{ij} if there is no budget constraints. From Theorem 3, $\sum_i \sum_j y^*_{ij} u_{ij} \ge \frac{1}{2} \sum_i \sum_j x^*_{ij} u_{ij}$. Now $\mu_i = \sum_j y^*_{ij} \le \sum_j x^*_{ij} \le b_i$ by budget constraint. Now apply Theorem 2 as in Theorem 1 to each advertiser separately. Hence, the expected revenue from advertiser i is at least $(1 - \frac{1}{e}) \sum_j y^*_{ij}$. Therefore, the total revenue is at least $(1 - \frac{1}{e}) \sum_i \sum_j y^*_{ij} \ge \frac{1}{2}(1 - \frac{1}{e}) \sum_i \sum_j x^*_{ij}$. So, overall, we get at least $\frac{1}{2}(1 - \frac{1}{e})$ of the optimal value of the expectation LP. Note that this is a gross overestimation because using conditional expectation on each customer may result in discarding some of the queries which would make it less likely for advertisers to hit their budget limit.

3.4 Proof of Lemma 2

Here we prove Lemma 2. We recall the statement of Lemma 2.

Lemma 2. *Consider the stochastic uniform knapsack problem as defined in Definition 2. Let O_o denote the random variable representing the expected revenue of Algorithm 3 for this problem (i.e. $O_o = E_1^C$, since C is the capacity of the knapsack)). Also define $O_e = \sum_j p_j v_j$. Assuming that $\sum_j p_j \le C$, the following always holds:*

$$\frac{1}{2} O_e \le E[O_o] \le O_e$$

Proof (Lemma 2). The upper bound is trivial. Clearly, no algorithm (offline or online) can get more than $p_j v_j$ revenue in expectation from each item j. So the total expected revenue is upper bounded by $O_e = \sum_j p_j v_j$. Next we prove the lower bound.

To prove the lower bound we first narrow down the instances that would give the smallest $\frac{E[O_o]}{O_e}$. The plan of the proof is as follows. First, we show that for each t if we replace all the items arriving at time t (i.e. all items in set T_t) with a single item with probability $p_t = \sum_{j \in T_t} p_j$ and value $v_t = \sum_{j \in T_t} v_j \frac{p_j}{p_t}$, we may only decrease $E[O_o]$ but O_e does not change. So this replacement may only decrease $\frac{E[O_o]}{O_e}$ and after the replacement, each partition only contains one item. So, WLOG, we only need to prove the lower bound for instances in which each partition contains one item. Next, we argue that if we scale all v_j's by a constant, both $E[O_o]$ and O_e are scaled by the same constant. So, WLOG, we assume that $O_e = 1$. Therefore, we only need to prove a lower bound on the following program:

$$\begin{array}{ll} \text{minimize.} & E[O_o] \\ \text{s.t.} & O_e \geq 1 \end{array}$$

We then consider a linear relaxation of the above program and prove a lower bound of $\frac{1}{2}$ on this relaxation which also implies a lower bound of $\frac{1}{2}$ for the original program. We prove this by constructing a feasible solution for the dual of this linear program that achieve a value of $\frac{1}{2}$.

In what follows, we explain each step of the proof in more detail:
First of all, we claim that if we replace all of the items arriving at time t (i.e., all items in T_t) with a single item with probability $p_t = \sum_{j\in T_t} p_j$ and value $v_t = \sum_{j\in T_t} v_j \frac{p_j}{p_t}$, then O_o may decrease but O_e is not affected. Let $E[O'_o]$ and O'_e respectively denote the result of making this replacement. The fact that O_e is not affected is trivial because $v_t p_t = \sum_{j\in T_t} p_j v_j$ so $O'_e = O_e$. Let ${E'}^r_t$ denote the expectation after replacing all items in T_{t^*} with a single item as explained. For all values of $t > t^*$ nothing is affected so ${E'}^r_k = E^r_k$. Consider what happens at $t = t^*$ when we make the replacement:

From (EXP) we have:

$$E^r_t = \sum_{j\in T_t} p_j \max(v_j + E^{r-1}_{t+1}, E^r_{t+1}) + (1 - \sum_{j\in T_t} p_j)E^r_{t+1}$$

for any convex function $f(\cdot)$ and nonnegative α_i's with $\sum_i \alpha_i = 1$ it always holds that $\sum_i \alpha_i f(x_i) \geq f(\sum_i \alpha_i x_i)$ (the Jensens inequality) and $\max(x + a, b)$ is a convex function of x so:

$$\begin{aligned} E^r_t &= p_t \sum_{j\in T_t} \frac{p_j}{p_t} \max(v_j + E^{r-1}_{t+1}, E^r_{t+1}) + (1 - p_t)E^r_{t+1} \\ &\geq p_t \max(\sum_{j\in T_t} \frac{p_j}{p_t} v_j + E^{r-1}_{t+1}, E^r_{t+1}) + (1 - p_t)E^r_{t+1} \\ &= p_t \max(v_t + E^{r-1}_{t+1}, E^r_{t+1}) + (1 - p_t)E^r_{t+1} \\ &= {E'}^r_t \end{aligned}$$

So we proved that ${E'}^r_t \leq E^r_t$ for $t = t^*$ and for all $r \in [1, C]$. Furthermore, notice that according to equation (EXP), for each t, E^r_{t-1} is an increasing function of E^r_t and E^{r-1}_t, since $\sum_{j\in T_t} p_j \leq 1$ (from mutual exclusivity of items arriving at time t). So if E^r_t and/or E^{r-1}_t decrease then E^r_{t-1} may only decrease. Therefore for all values of $t \leq t^*$ we can argue that ${E'}^r_t \leq E^r_t$ and in particular $E[O'_o] = {E'}^C_1 \leq E^C_1 = E[O_o]$. That means the replacement may only decrease the expected revenue of our algorithm. So if we replace all the items in each T_t with a single item as explained above one by one we get an instance in which each partition only contains one item and with a possibly lower expected revenue from our algorithm. Therefore, WLOG, it is enough to prove a lower bound for the case where each partition contains one item.

Since scaling all v_j's by a constant scales both $E[O_o]$ and O_e by the same constant, we can scale all v_j's so that $O_e = 1$. So, WLOG, we only need to prove the lower bound for cases where $O_e = 1$. Now, we argue that the optimal value of the following program gives a lower bound on $E[O_o]$. Therefore, we only need to prove the optimal value of this program is bounded below by $\frac{1}{2}$.

$$\begin{array}{ll} \text{minimize.} & E[O_o] \\ \text{s.t.} & O_e \geq 1 \end{array}$$

We now rewrite the previous program as the following linear program with variables $\boldsymbol{E}^r_t$ and $\boldsymbol{v}_t$ (with $t \in [u]$ and $r \in [C]$ by using the definition of E^r_t from (EXP). Note that $E[O_o] = E^C_1$. Also note that, in the following, we address each item by the index of the partition to which it belongs, since there is only one item in each partition.

$$\begin{array}{ll}
\text{minimize.} & \boldsymbol{E}_1^C \\
\forall t \in [1, u-1], \forall r \in [1, C]: & \boldsymbol{E}_t^r \geq p_t(\boldsymbol{v}_t + \boldsymbol{E}_{t+1}^{r-1}) + (1-p_t)\boldsymbol{E}_{t+1}^r \\
\forall t \in [1, u-1], \forall r \in [1, C]: & \boldsymbol{E}_t^r \geq \boldsymbol{E}_{t+1}^r \\
\forall r \in [1, C]: & \boldsymbol{E}_u^r \geq p_u \boldsymbol{v}_u \\
 & \sum_t p_t \boldsymbol{v}_t \geq 1 \\
\forall r \in [1, C]: & \boldsymbol{v}_t \geq 0, \quad \boldsymbol{E}_t^r \geq 0 \\
\forall t \in [1, u] & \boldsymbol{E}_t^0 = 0
\end{array}$$

Notice that any feasible assignment for the original program satisfies all the constraints of the above program and hence is also a feasible assignment for the above program. Therefore its optimal value is a lower bound for the optimal value of the original program.

The above linear program is still not quite easy to analyze, so we consider a looser relaxation as we explain next. First, it is not hard to show that E_t^r as defined in (EXP) has decreasing marginal value in r which implies $E_t^{r-1} \geq \frac{r-1}{r} E_t^r$ (This can be proved by induction on t with the base case being $t = u$ and then proving for smaller t's. We will prove this formally later). Combining this with the definition of E_t^r from (EXP), we get the following inequality:

$$\begin{aligned}
E_t^r &= p_t \max(v_t + E_{t+1}^{r-1}, E_{t+1}^r) + (1-p_t)E_{t+1}^r \\
&= \max(p_t v_t + p_t E_{t+1}^{r-1} + (1-p_t)E_{t+1}^r, E_{t+1}^r) \\
&\geq \max(p_t(v_t + \frac{r-1}{r} E_{t+1}^r) + (1-p_t)E_{t+1}^r, E_{t+1}^r) \\
&= \max(p_t v_t + (1 - \frac{p_t}{r})E_{t+1}^r, E_{t+1}^r)
\end{aligned}$$

Using the above, we write a more relaxed linear program.

$$\begin{array}{lll}
\text{minimize.} & \boldsymbol{E}_1^C & \\
\forall t \in [1, u-1], \forall r \in [1, C]: & \boldsymbol{E}_t^r - p_t \boldsymbol{v}_t - (1 - \frac{p_t}{r})E_{t+1}^r \geq 0 & (3.1) \\
\forall r \in [1, C]: & \boldsymbol{E}_u^r - p_u v_u \geq 0 & (3.2) \\
\forall t \in [1, u-1], \forall r \in [1, C]: & \boldsymbol{E}_t^r - \boldsymbol{E}_{t+1}^r \geq 0 & (3.3) \\
 & \sum_{t=1}^u p_t \boldsymbol{v}_t \geq 1 & (3.4) \\
\forall t \in [1, u], \forall r \in [1, C]: & \boldsymbol{v}_t \geq 0, \quad \boldsymbol{E}_t^r \geq 0 & (3.5) \\
\forall t \in [1, u] & \boldsymbol{E}_t^0 = 0 & (3.6)
\end{array}$$

We now further relax the above linear program, and only keep the constraints for $r = C$.

$$\begin{array}{lll}
\text{minimize.} & \boldsymbol{E}_1^C & \\
\forall t \in [u-1]: & \boldsymbol{E}_t^C - p_t \boldsymbol{v}_t - (1 - \frac{p_t}{r})E_{t+1}^C \geq 0 & (\boldsymbol{\alpha}_t) \\
 & \boldsymbol{E}_u^C - p_u v_u \geq 0 & (\boldsymbol{\alpha_u}) \\
\forall t \in [u-1]: & \boldsymbol{E}_t^C - \boldsymbol{E}_{t+1}^C \geq 0 & (\boldsymbol{\beta}_t) \\
 & \sum_{t=1}^u p_t \boldsymbol{v}_t \geq 1 & (\gamma) \\
 & \boldsymbol{v}_t \geq 0, \quad \boldsymbol{E}_t^C \geq 0 &
\end{array}$$

Next, we show that the optimal value of the above program is bounded below by $\frac{1}{2}$ which implies that the optimal value of the original program is also bounded below by $\frac{1}{2}$ and that completes the proof. To do this, we present a feasible assignment for the dual program that obtains an objective value of at least $\frac{1}{2}$. Note that the objective value of any feasible assignment for the dual program gives a lower bound on the optimal value of the primal program. The following is the dual program:

$$
\begin{array}{lll}
\text{maximize.} & \boldsymbol{\gamma} & \\
\forall t \in [u]: & \boldsymbol{\gamma} p_t - \boldsymbol{\alpha}_t p_t \leq 0 & (\boldsymbol{v}_t) \\
 & \boldsymbol{\alpha}_1 + \boldsymbol{\beta}_1 \leq 1 & (\boldsymbol{E}_1^C) \\
\forall t \in [2 \cdots u-1]: & \boldsymbol{\alpha}_t + \boldsymbol{\beta}_t - (1 - \frac{p_{t-1}}{C})\boldsymbol{\alpha}_{t-1} - \boldsymbol{\beta}_{t-1} \leq 0 & (\boldsymbol{E}_t^C) \\
 & \boldsymbol{\alpha}_u - (1 - \frac{p_{u-1}}{C})\boldsymbol{\alpha}_{u-1} - \boldsymbol{\beta}_{u-1} \leq 0 & (\boldsymbol{E}_u^C) \\
 & \boldsymbol{\alpha}_t \geq 0, \quad \boldsymbol{\beta}_t \geq 0, \quad \boldsymbol{\gamma} \geq 0 &
\end{array}
$$

Now, suppose we set all $\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} - \frac{p_{t-1}}{C}\boldsymbol{\gamma}$ for all t except $\boldsymbol{\beta}_1 = 1 - \boldsymbol{\gamma}$. And, we set all $\boldsymbol{\alpha}_t = \boldsymbol{\gamma}$ except $\boldsymbol{\alpha}_u = \boldsymbol{\beta}_u + \boldsymbol{\gamma}$. From this assignment, we get $\boldsymbol{\beta}_t = 1 - \boldsymbol{\gamma} - \boldsymbol{\gamma}\sum_{k=1}^{t-1}\frac{p_k}{C}$. Observe that we get a feasible solution as long as all $\boldsymbol{\beta}_t$'s resulting from this assignment are non-negative. Furthermore, it is easy to see that $\boldsymbol{\beta}_t > 1 - \gamma - \gamma\frac{\sum_{k=1}^{u} p_k}{C} = 1 - 2\boldsymbol{\gamma}$. Therefore, for $\boldsymbol{\gamma} = \frac{1}{2}$, all $\boldsymbol{\beta}_t$'s are non-negative and we always get a feasible solution for the dual with an objective value of $\frac{1}{2}$ which completes the main proof. Next, we present the proof of our earlier claim that $E_t^{r-1} \geq \frac{r-1}{r}E_t^r$.

We now prove that $E_t^r \geq \frac{r}{r+1}E_t^{r+1}$ by induction on t with the base case being $t = u$ which is trivially true because $E_u^r = p_u v_u$ for all $r \geq 1$. Next we assume that our claim holds for $t+1$ and all values of r. We then prove it for t and all values of r as follows:

$$
\begin{aligned}
E_t^r &= p_t \max(v_t + E_{t+1}^{r-1}, E_{t+1}^r) + (1-p_t)E_{t+1}^r \\
&= \max(p_t(v_t + E_{t+1}^{r-1}) + (1-p_t)E_{t+1}^r, E_{t+1}^r)
\end{aligned}
$$

Observe that $\max(a,b) \geq \max((1-\epsilon)a + \epsilon b, b)$ *for all* $\epsilon \in [0,1]$ *so:*

$$
\begin{aligned}
E_t^r &\geq \max((1-\epsilon)[p_t(v_t + E_{t+1}^{r-1}) + (1-p_t)E_{t+1}^r] \\
&\quad + \epsilon E_{t+1}^r, E_{t+1}^r) \\
&= \max((1-\epsilon)p_t(v_t + E_{t+1}^{r-1} + \frac{\epsilon}{1-\epsilon}E_{t+1}^r) \\
&\quad + (1-p_t)E_{t+1}^r, E_{t+1}^r)
\end{aligned}
$$

Now by applying the induction hypothesis on E_{t+1}^{r-1} *and* E_{t+1}^r *and setting* $\epsilon = \frac{1}{r+1}$*:*

$$
\begin{aligned}
E_t^r &\geq \max(\frac{r}{r+1}p_t[v_t + \frac{r-1}{r}E_{t+1}^r + \frac{1}{r}E_{t+1}^r] \\
&\quad + (1-p_t)\frac{r}{r+1}E_{t+1}^{r+1}, \frac{r}{r+1}E_{t+1}^{r+1}) \\
&= \frac{r}{r+1}\max(p_t[v_t + E_{t+1}^r] + (1-p_t)E_{t+1}^{r+1}, E_{t+1}^{r+1}) \\
&= \frac{r}{r+1}E_t^{r+1}
\end{aligned}
$$

So we proved that $E_t^r \geq \frac{r}{r+1}E_t^{r+1}$ and that completes the proof. □

References

1. M. Agrawal. Overview of mobile advertising. `http://www.telecomcircle.com/2009/11/overview-of-mobile-advertising`.
2. N. Andelman and Y. Mansour. Auctions with budget constraints. In *In 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, 2004.
3. Y. Azar, B. Birnbaum, A. R. Karlin, C. Mathieu, and C. T. Nguyen. Improved approximation algorithms for budgeted allocations. *ICALP '08*. 186-197, 2008.

4. D. Chakrabarty and G. Goel. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. *SIAM J. Comp.* p2189, 2010.
5. N. R. Devenur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. *EC '09*. 71-78, 2009.
6. J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. *FOCS '09*. 117-126, 2009.
7. G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. *SODA '08*. 982-991, 2008.
8. M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. *EC '04*. 71-80, 2004.
9. M. T. Hajiaghayi, R. D. Kleinberg, and T. Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI*, 2007.
10. K. Joag-Dev and F. Proschan. Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295, 1983.
11. C. Karande, A. Mehta, and P. Tripathi. Online bipartite matching with unknown distributions. In *STOC*, 2011.
12. R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. *STOC '90*. 352-358, 1990.
13. D. P. Kennedy. Prophet-type inequalities for multi-choice optimal stopping. In *Stoch. Proc. Applic.*, 1987.
14. U. Krengel and L. Sucheston. Semiamarts and finite values. *Bull. Am. Math. Soc.*, 1977.
15. U. Krengel and L. Sucheston. On semiamarts, amarts, and processes with finite value. *In Kuelbs, J., ed., Probability on Banach Spaces.*, 1978.
16. S. Lahaie. An analysis of alternative slot auction designs for sponsored search. *EC '06*. 218-227, 2006.
17. M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing lps. In *STOC*, 2011.
18. V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. *SODA '11*, 2011.
19. A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54, 2007.
20. A. Srinivasan. Budgeted allocations in the full-information setting. *APPROX '08*. 247-253, 2008.
21. S. Wang, J. Min, and B. K. Yi. Location based services for mobiles: Technologies and standards. *ICC '08*, 2008.
22. F. Zahradnik. Garmin to offer free, advertising-supported traffic detection and avoidance. `http://gps.about.com/b/2008/09/15/garmin-to-offer-free-advertising-supported-traffic-detection-and-avoidance.htm`.

A AdCell Policies

AdCell strategy presents new modeling and algorithmic challenges for making it work efficiently. We are introducing AdCell as a new concept in Internet Marketing/Advertising via Cell Phones. Consider a WSP which provides mobile services to cell phone users. The WSP can easily use the location of the user to target the advertisements by demographic, interests, Designated Market Area[6], mobile device and carrier which improves the ad experience of customers and advertising of *Advertiser Companies*. In this section we discuss the possible mechanisms for improving advertising and motivating customers to participate.

A.1 Customers Policy

The WSP may provide different AdCell plans. Each plan offers discounts for agreeing to receive advertisement via SMS and based on the customer's geographical location. For example 5% discount for receiving 50 ads every month or 10% discount for receiving 100 ads every month.

In addition we may have a *profile form* for each customer (e.g. as an online form). Customers are encouraged to fill this form and based on this information, we would be able to design *customer types* which can provide additional information for predicting if an ad would be useful for that person. For example we may ask about gender, age, what kind of food shops they like, etc in the form. This may help us improve the ad experience of each individual person, specially comparing to the simple AdWords Advertising in search engines which are indifferent to the customer who is searching the keyword.

A.2 Scoring Policy

Clearly the wireless provider should encourage the customers for participating in AdCell service. Other than the discount for singing in the plans, we may improve the ratio of successful ads by implementing a scoring policy such that a customer with high score may get extra discounts. The idea is that when a customer visits a store -

[6] Or *Media Market* is a region where the population can receive the same (or similar) television and radio station offerings and may also include other types of media.

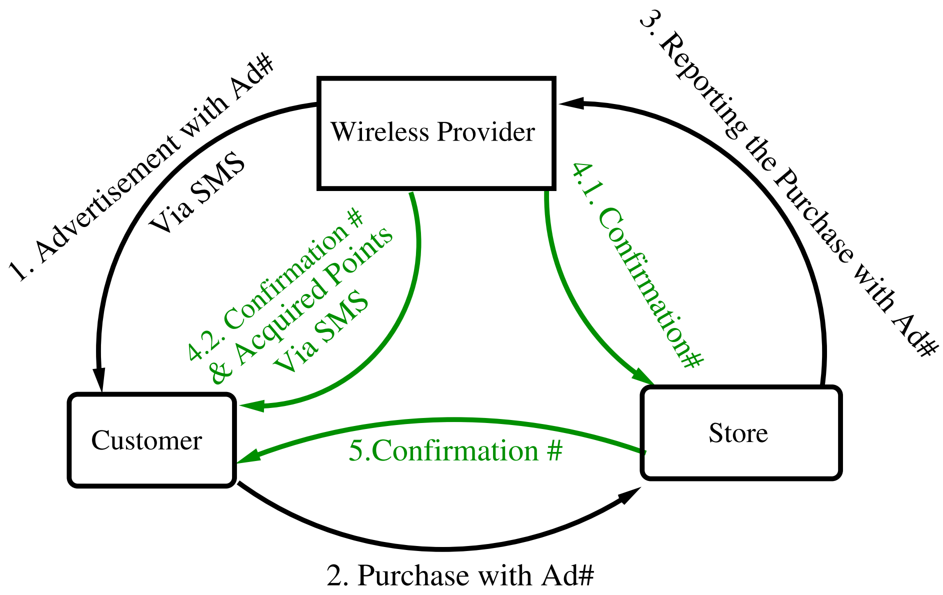

Fig. A.1. A Purchase Scenario. Numbers show the sequence of events.

due to an advertisement - she should get some points (and since the ad was successful, the WSP would charge the store an additional fee). The acquired point may depend on the plan the customer selects or how much she spends at that participating store. In order for the customer to get the points, the store where she visits must report it to the WSP.

If a store reports a visit then that store will have to pay an additional fee. Thus to be able to enforce the scoring policy, each advertisement should contain an *ad number* that encodes *the user id, date (or the range over which the ad is valid), and store id*. A scenario for a purchase based on AdCell ads is shown in Figure A.1. The customer must show this *one-time-usable* ad number to the store, the store calls the WSP (or does it by Internet) and provides this ad number. The WSP upon receiving the ad number will confirm the call by the store, by providing a confirmation number and the store will give this number to the customer. Also the WSP will send the same confirmation number and a notification about the acquired points via SMS to the customer. Specific parts of the confirmation number may show a customer-specific id (only provided and known to the customer) and thus the customer can verify the confirmation number by checking this specific part; or by comparing it to the number in the confirmation SMS from WSP.

In case of a fraud, customers can always submit a complain providing the ad number, receipt from the store and the approximate time of the visit. The WSP can check the location and the receipt to verify that the customer is telling the truth and then take an appropriate action. Stores get a score based on their reputation. Frauds will decrease the reputation significantly and the ratio of successful ads will increase the reputation. If an store's score becomes low, the company either stops giving them advertisement slots or charges them much more just for showing the ad.