Online Allocation of Display Advertisements Subject to Advanced Sales Contracts

Saeed Alaei
University of Maryland
A.V. Williams Building
College Park, MD 20740.
saeed@cs.umd.edu

Esteban Arcaute
Stanford University
496 Lomita Mall
Stanford, CA 94305.
arcaute@stanford.edu

Samir Khuller
University of Maryland
A.V. Williams Bldg., Room 3369
College Park, MD 20740.
samir@cs.umd.edu

Wenjing Ma
Yahoo! Inc.
701 First Ave
Sunnyvale, CA 94089.
wenjingm@yahoo-inc.com

Azarakhsh Malekian
University of Maryland
A.V. Williams Building
College Park, MD 20740.
malekian@cs.umd.edu

John Tomlin
Yahoo! Research
701 First Ave
Sunnyvale, CA 94089.
tomlin@yahoo-inc.com

ABSTRACT

In this paper we propose a utility model that accounts for both sales and branding advertisers. We first study the computational complexity of optimization problems related to both online and offline allocation of display advertisements. Next, we focus on a particular instance of the online allocation problem, and design a simple online algorithm with provable approximation guarantees. Our algorithm is near optimal as is shown by a matching lower bound. Finally, we report on experiments to establish actual case behavior on some real datasets, with encouraging results.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity - *general*; J.m [**Computer Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation, Modeling

Keywords

Display Advertising, Online algorithms, Simulation, Optimization

1. INTRODUCTION

The current system for allocating and pricing display advertisements relies on two separate markets. On the one hand, *contracts* are set between the publisher and advertisers interested in large volumes of impressions. This is done through bilateral bargaining between the advertisers and the sales force of the publisher. On the other hand, the remaining inventory is sold on a per-impression basis through an *auction* mechanism.

There are several justifications for the co-existence of such different products in the same market place. At first sight, the sale of *advance purchase* contracts seems to indicate that the publisher seeks to protect itself against uncertain demand [7]. While simple and appealing, this explanation fails to account for the ease with which some contracts are sold. Assuming the publisher *only* sells advance purchase contracts for products with uncertain demand, and price them accordingly, one does not expect impressions subject to uncertain demand to sell-out early on contracts alone.

Further inspection of the *buyers* of display advertisement impressions reveals a different picture. Alongside regular retailers like Macy's or Nordstrom, one can see brands with no point of sale such as Coca-Cola or Intel. While it is reasonable to assume a retailer has sufficient information to determine the value of *an unique impression*, as an impression can be linked to a sale and thus can be assigned an expected value [8], it is unlikely for a brand like Coca-Cola to be able to measure the value of an impression outside the scope of a *campaign*.

More specifically, we can assume there are two extreme advertisers' profiles. The first extreme can be characterized by retailers like Nordstrom, that can be thought of as *sales driven* as they seek to *transform impressions into sales*. The second extreme can be characterized by the likes of Coca-Cola, that can be thought of as *branding driven*, as they seek to create *brand equity* [10] from their advertisement campaigns.

The utility function of sales driven advertisers can be modeled assuming they view impressions as commodities. Modeling the utility function of branding driven advertisers requires *bundling* more than one impression together. Thus there is an implicit *complementarity* between impressions (or groups of impressions) for branding driven advertisers, alongside the inherent substitutability of impressions.

The contributions of the present paper are as follows. We first propose a utility model that accounts for both sales and branding advertisers. Next, we study the computational complexity of optimization problems related to both online and offline allocation of display advertisements. We then focus on a particular instance of the online allocation problem, and design a simple online algorithm with provable approximation guarantees. Our algorithm is near op-

ADKDD'09, June 28, 2009, Paris, France.

timal as is shown by a matching lower bound. Finally, we report on experiments to establish actual case behavior on some real datasets, with encouraging results.

It is important to note that even highly idealized versions of the problems we seek to model are NP-hard. Assuming we have full information about the supply of impressions and the advertisers' types, we show that the problem of optimally allocating impressions to advertisers is NP-hard. We explicitly reduce problems such as the Santa Claus Problem [2], which is known to be NP-hard (in fact, the first non-trivial approximation to the Santa Claus problem was only recently given by Asadpour and Saberi in [1]), to our problem. In the more restricted case where only one type of impression is available, we show that Resource Allocation [3] can be reduced to our problem, which shows that our problem is NP-hard even in the simplest of settings.

We then consider the online case where advertisers (and thus contracts) arrive online. Is such setting, we show that even the case where one impression is available per round is hard to approximate [11]. Inspired by the results from [11], we consider a setting where contracts can be dropped without penalties (it is important to note that the - offline- problem where penalties are generated upon dropping a contract has been studied in [5]). There we present an online algorithm with approximation ratio linear in the number of contracts, and provide a matching lower bound. Note that, as in [11], we consider a *deterministic* algorithm. Recently, in [6] and [4], the authors considered the use of *randomized* algorithms in the online setting.

We simulate the performance of our algorithm using 2007 data derived from Yahoo!'s display advertisement business. We conclude that the instances encountered allow our algorithm to perform with little performance loss.

Organization of the Paper

The rest of the paper is organized as follows. In Section 2 we present a simple utility model considered for both advertisers' types. In this model only one type of impression is assumed to exist. We deferred the presentation of the full model to Appendix A as most of the paper deals with the case where only one type of impression is available. In Section 3 we explore the computational complexity of problems related to the allocation of display advertisements with agents' utilities given by our model. In Subsection 3.1 we show that in a general setting, the problem of maximizing the utility of the least happy advertiser is NP-hard, even when all contracts have unit duration. In such formulation of the problem, the hardness comes from the multiplicity of impression types, and from the assumption that they can be considered as substitutes.

Next, in Subsection 3.2, we explore the computational complexity associated with contracts having different durations. To do so, we assume that there is only one impression type, thus avoiding the computational complexity of Subsection 3.1. We there show that it is NP-hard to maximize social welfare (or revenue) even if all information about supply and advertisers types' is known.

In Section 4 we present an online setting motivated by the procedure followed when advanced sales contracts are booked. In Subsection 4.1 We first show that, in general, it is impossible to approximate social welfare to any constant in the general case. Next, we consider the case where contracts can be dropped without penalties. There we extend the results found in [11] from one impression per round to an arbitrary number of impressions per round. We provide an approximation algorithm together with a matching lower bound.

Finally, in Subsection 4.3 we test the performance of our algorithm against data extracted from Yahoo!'s display advertisement business.

Summary of Contributions

We provide a utility model for display advertisement that encompasses both branding and sales advertisement. The model is rich enough to express settings where multiple types of impressions are available, and allows to model both substitution and temporal complementarity effects between impressions.

We prove that, for the case of multiple impression types, it is NP-hard to compute a max-min fair allocation of impressions. We then prove that the problem reduces to a linear program if the impressions are assumed to be divisible. In that same setting, even when one type of impression is available, we prove it is NP-hard to maximize social welfare. The rest of the paper deals with this setting.

We then assume contracts arrive online and provide a lower bound on the competitive ratio of any online algorithm. We provide a simple greedy algorithm that is optimal (up to a constant factor). Since the lower bound on the competitive ratio is quite pessimistic, we simulate the performance of our greedy algorithm on actual data from Yahoo!'s display advertisement business. We find that our algorithm performs very well in practice as the observed performance loss is very small to that predicted in the lower bound.

2. UTILITY MODEL

We assume that advertisers work over business periods spanning $T > 0$ individual rounds. One can think of the business period as the time horizon advertisers' decide to maximize their utility with respect to, and of rounds as the smallest quanta of time over which an advertiser might have a constraint in defining its utility. As an example, a typical business period is one quarter (four months). For branding purposes, one can assume that an advertiser is not willing to see a day pass without an impression being allocated to him/her. Thus, in this example, a round would be defined as a day. More generally, one can think of the business period as defining the *long term objectives* of the advertiser, and of a round as providing *short term utility constraints* for the advertisers.

In all that follows we use the following notation. Each round is labeled by $t \in \{1, \ldots, T\}$. We assume that during round t, n_t identical items will be available for sale. We note

$$\mathbf{S} = \prod_{t=1}^{T} \{0, \ldots, n_t\}$$

where $s \in \mathbf{S}$ represents a possible allocation of impressions to a given advertiser. Let $N = \sum_{t=1}^{T} n_t$ be the total supply over the business period.

We index advertisers by $k \in \{1, \ldots, K\}$ and note them A_k. An *allocation* $\mathcal{S} = \{s_1, \ldots, s_K\}$ is a k-partition of the set impressions such that,

- for all $k \in \{1, \ldots, K\}$, $s_k \in \mathbf{S}$; and
- $\sum_k s_k \leq (n_1, \ldots, n_T)$.

The vector s_k represents the set of impressions allocated to advertiser A_k. The second condition states that we do not allocate more impressions during a given round than available.

We consider a general utility model for advertisers. Advertiser A_k is fully characterized by a 5-tuple $(t_k, \tau_k, \delta_k, b_k, V_k)$ where

- $t_k \in \{1, \ldots, T\}$ is the *start date* advertiser A_k is interested in; and
- $\tau_k \in \{1, \ldots, T\}$ is the *duration* advertiser A_k is interested in; and

- $\delta_k \in \{1, \ldots, n\}$ is the *minimum impressions per round* advertiser A_k seeks to receive starting at t_k and for τ_k rounds; and

- $b_k \in \mathbb{R}^+$ is the *utility* drawn from receiving δ_k impressions per round during τ_k consecutive rounds; and

- V_k is the *additional value* function

$$V_k : \{0, 1, \ldots, N\} \rightarrow \mathbb{R}$$
$$q \mapsto V_k(q)$$

where $V_k(q)$ is the value drawn by advertiser A_k of being allocated q additional impressions *given* that it is guaranteed to receive δ_k impressions per round during τ_k consecutive rounds. Note that $V_k(q) < 0$ can model ad-fatigue.

Given an allocation $s_k \in \mathbf{S}$, the utility to agent A_k is

$$U(s_k; A_k) = b_k + V_k(|s_k| - \delta_k \tau_k) \qquad (1)$$

provided $\forall t \in [t_k, t_k + \tau_k - 1],\ s_k(t) \geq \delta_k$. Otherwise the utility to agent A_k is assumed to be nil.

The previous utility function simply says that, if agent A_k is *not* given an allocation where it receives at least δ_k items per round during rounds in $[t_k, t_k + \tau_k - 1]$, then A_k's utility is nil. This accounts for branding activities linked to display advertising.

In Examples 1 and 2 we use the utility function form Equation 1 to model both branding and sales advertisers' utilities.

Example 1 (Sales Advertisers)
As noted in Section 1, sales driven advertisers view impressions as commodities. Thus if we set $t_k = \tau_k = 1$ and $\delta_k = 0$, we conclude that such advertiser, impressions across rounds are perfect substitutes, and thus the utility function obtained models the setting where impressions are commodities (here the value advertiser A_k has for q impressions is given by $V_k(q)$)

Example 2 (Branding Advertisers)
An extreme branding example would be to consider $t_k = 1$, $\tau_k = T$ and $\delta_k \leq \min_t n_t$. Thus, in this example, the advertiser does not value an allocation unless it is guaranteed at least δ_k impressions per round during *every round*!

In order to best serve advertisers interested in branding activities, we assume the publisher proposes *advanced sales contracts* as defined bellow.

Definition 1 ((Advanced Sales) Contracts)
A contract $\Gamma(t, \tau, \delta, p)$ is a guarantee that the publisher will provide δ impressions per round over τ consecutive rounds starting at round t at a price of p. We call t the start of the contract, τ the duration, δ the quantity and p the price.

It is important to note that an advanced sales contract is a *guarantee* of delivery of a given set of impressions to a given advertiser during a given period of time. As such, we will consider (unless explicitly stated) a contract to be an *obligation*, and thus will not deal with penalties associated with under-delivery. To do so, we assume that the supply of impressions per round is known.

In order to deal exclusively with contracts, we assume that spot sales of impressions are represented by a contract (there $t = \tau = 1$ and $\delta = 1$).

3. COMPUTATIONAL COMPLEXITY

In this section we explore the computational complexity of two problems associated with the sale and allocation of display advertisements. We consider two main components necessary to fully model the utility of advertisers interested in display advertisements. The first is the inherent substitutability of impression types. The second is induced by the complementarity of impressions across time due to branding effects.

In Subsection 3.1, we show that substitutability across different impressions types makes the problem NP-hard even in the absence of complementarity effects. We do so by reducing the Santa Claus problem to our setting. Next we show in Subsection 3.2, that complementarity of impressions across time (as induced by branding) makes the problem NP-hard even in the presence of a unique type of impressions. We do so by reducing the Resource Allocation problem to our setting.

3.1 Multiple Impressions - Santa Claus

In this subsection we present a well known NP-hard problem, the Santa Claus problem. We then point out how it is included as a special case of our display advertisement allocation problem.

Santa Claus seeks to allocate each of n distinct toys, call them $g_1, \ldots, g_n$, to K distinct children, call them $C_1, \ldots, C_K$. Each child C_k has a personal value of $u_{kj} \geq 0$ for the toy g_j. We assume that if a child receives more than one toy, then the value of the bundle it receives is the sum of the values of the toys included in the bundle.

Santa Claus seeks to allocate all n toys in order to make the *least lucky* child as happy as possible. In other words, Santa Claus seeks to maximize the minimum utility across all children. Such and allocation is usually referred to as a *max-min fair allocation*.

Formally, given an allocation vector $S \in \{1, \ldots, K\}^n$, where $S_j = k$ implies that toy g_j was given to child C_k. Let $\mathcal{J}_k(S) = \{j :\ S_j = k\}$ be the set of indices of the goods given to child C_k under allocation S. Then an allocation S_{SC} that solves the Santa Claus problem is such that

$$S_{SC} \in \operatorname*{arg\,max}_{S' \in \{1,\ldots,K\}^n} \left[\min_{k \in \{1,\ldots,K\}} \sum_{j \in \mathcal{J}_k(S')} u_{kj} \right]$$

In our setting, assume that we have just one round, n different types of impressions, and only one impression per type. Assume that all advertisers are "sales advertisers" as defined in Example 1, but that their utility is given by Equation 17.

If we are interested in making the least lucky advertiser as happy as possible, it is clear that the setting described above is identical to the Santa Claus problem. We now need to provide a justification about using max-min fair allocation as a solution concept.

There are two important aspects not modeled in our current setting. First, advertisers transact with the publisher *repeatedly*. This implies that the objective maximized by advertisers can depend on past transactions. Further, we do not consider competition from other publishers. By considering max-min fair allocations, we are trying to account for maintaining "good business relationships" with advertisers. With this in mind, one can conceivably see how we can indirectly incorporate those two important aspects into our model with the proper choice of objective function.

3.2 Single Impression - Resource Allocation

In Subsection 3.1 we proved that our setting includes NP-had problems that are difficult to approximate. Upon further inspection of the reduction used, it is clear that we did not used what makes our problem novel, namely the utility model used to account for branding effects.

In this subsection we restrict ourselves to a setting not including the Santa Claus problem as a special case. We show that considering branding effects when only a unique type of impression is available is of individual interest. We first show that, in this setting, the max-min fair allocation problem is easy to solve. We then show that maximizing social welfare (or revenue) is NP-hard.

In all that follows, we assume that the number of impressions per round is sufficiently large so that impressions can be considered a divisible good. This is needed as one can easily reduce knapsack to our setting. Assume K advertisers, named $A_1, \ldots, A_K$ are interested in buying impressions. We further assume that all attributes $(t_k, \tau_k, \delta_k, b_k, V_k)$ for advertiser A_k are known. We assume that $V_k(q) = v_k q$ is a linear function in q (possibly equal to zero). In order to simplify notation, we assume without loss of generality that during each round n impressions are available.

3.2.1 Problem Formulation - LP

Let $s_{t,k} \in [0, n]$ denote the number of impressions (possibly fractional) allocated to advertiser A_k during round $t \in \{1, \ldots, T\}$. Let us first describe the supply constraints:

$$\forall t \in \{1, \ldots, T\}, \ \sum_{k=1}^{K} s_{t,k} \leq n. \tag{2}$$

Next, we describe the branding constraint associated with advertiser A_k:

$$\forall t \in \{t_k, \ldots, t_k + \tau_k - 1\}, \ s_{t,k} \geq \delta_k. \tag{3}$$

Thus we can see that all constraints associated with our utility function for advertisers are simply linear constraints. In the next Subsubsection, we show how the max-min fair allocation problem can be modeled as an LP, and thus is easy to solve.

3.2.2 Max-Min Fair Allocation Problem

Consider adding the following constraints to those represented by Equations 2 and 3.

$$\forall k \in \{1, \ldots, K\}, \ b_k + v_k \left[\sum_{t=t_k}^{t_k+\tau_k-1} (s_{t,k} - \delta_k) + \sum_{t=t_k+\tau_k} s_{t,k} \right] \geq y \tag{4}$$

and let the objective be to maximize $y > 0$. First, note that if the supply is not enough to satisfy all contracts, then the LP is not feasible. In that case, set the objective to zero as the utility of the least happy advertiser is zero. In all other cases, the LP we formulated maximizes y, which, according to the inequalities from Equation 4, is equal to the utility of the least lucky advertiser.

Since the number of constraints is $T + k + k$, this problem can be solved efficiently.

3.2.3 A Difficult Problem - Resource Allocation

In Subsubsection 3.2.2 we showed how the problem of finding a max-min fair allocation is equivalent to solving an LP. Here we argue that even in such idealized conditions (namely impressions being fractionally allocated, and only one type of impressions) a natural problem related to *admission control* is NP-hard.

What makes the max-min fair allocation problem easy to solve is the assumption that *all advertisers must meet their branding constraints*. Even one advertiser not meeting its branding constraint implies a max-min fair allocation value of zero. If the LP is not feasible (i.e. if supply is not enough to serve all advertisers), we can ask what is the maximum social welfare(or revenue) we can obtain subject to supply constraints?

It turns out that maximizing social welfare in our setting is NP-hard. A special instance of our problem, called "Resource Allocation", can be approximated to within a constant factor $(1/2 - \varepsilon)$ efficiently (see [3]). The setting for Resource Allocation is such that, for all k, $V_k = 0$. In other words, if advertisers are only interested in satisfying their branding constraints, and we are interested in maximizing social welfare (or revenue), our problem is identical to Resource Allocation.

A very interesting result from [3] is the existence of a deterministic algorithm with approximation ratio of 3 that takes $O(K^2 \log^2(K))$.

4. ONLINE SETTING

In Section 3 we implicitly assumed that all advertisers arrived at the same time, and thus that all contract data was available simultaneously. In practice contracts are negotiated at different times, and the decision to accept or reject a proposed contract is done online. We consider an online variant of the setting from Subsubsection 3.2.3.

In Subsection 4.1 we show how, if contracts can be dropped with no penalty and no prior is known on the distribution of the arriving contracts (i.e. in a fully adversary setting), then even the case of allocating one impression per round cannot be approximated to within any constant ratio. In a more restricted setting where impressions are valued similarly by all advertisers, using results from [11] we show that an approximation ratio of 4 is optimal.

Motivated by the setting and results form [11], in Subsection 4.2 we consider a setting where n resources are available and contracts arrive online. We propose a greedy algorithm that, for a class of valuations of the contracts similar to that studied in [11], achieves a competitive ratio of at most $4n$, thus generalizing the result from [11] (where $n = 1$ and the competitive ratio obtained was 4). We also present a matching lower bound for the competitive ratio of n, thus showing that our algorithm is almost optimal.

In Subsection 4.3 we simulate the performance of our algorithm against real data. Surprisingly, we find that our algorithm performs within only a few percentage points of the optimal omniscient offline algorithm.

4.1 Preliminary Results

Here we present the setting from [11] and some of its results. Assume we are interested in the following problem. Advertisers arrive online and request a contract. Upon arrival, advertiser k reveals all its information $(t_k, \tau_k, q_k, b_k, V_k)$ and requests a contract. We assume that advertiser A_k arrives at time t_k. The publisher then decides on the spot to accept or reject the contract.

In the setting considered in [11] it is assumed that

- $n = 1$ (only one impression per round is available)
- $q_k = 1$ (all of the supply is exhausted by one advertiser at any given time)
- $V_k = 0$ (advertisers are only interested in branding)
- payment is received once the contract is fully completed.
- contracts can be dropped without penalty (i.e. the publisher can accept a contract and drop it later if a better one arrives)

The goal is to maximize social welfare (or revenue) under *adversarial arrival* of advertisers. It is clear that no online algorithm can have a bounded approximation ratio without further assumptions, thus there is an extra assumption that links the duration of the contract τ to its value b (which look a lot like convexity).

Under such assumptions the authors give a 4 approximation algorithm. It is interesting to note that the approximation ratio is tight, and that the algorithm is very simple (greedy algorithm on contract value).

4.2 Online Algorithm

In [11] the assumption is that $n_t = 1, 1 \leq t \leq T$. However it is not the correct assumption for our problem. The basic model is the same: Advertisers arrive online and request a contract. Upon arrival, advertiser k reveals all its information $(t_k, \tau_k, q_k, b_k, V_k)$ and requests a contract. We assume that advertiser A_k arrives at time t_k. The publisher then decides on the spot to accept or reject the contract.

However the constraints in our problem, should be relaxed as follows:

- $n \geq 1$ we can have more than one resource per day but it is the same for all rounds.
- $q_k \leq n$ and it is possible to have more than one active advertiser per round however their total demand should not exceed n.
- $V_k = 0$ (advertisers are only interested in branding)
- payment is received once the contract is fully completed.
- contracts can be dropped without penalty (i.e. the publisher can accept a contract and drop it later if a better one arrives)

The goal is to maximize social welfare (or revenue) under *adversarial arrival* of advertisers. We call this problem with the properties defined above *Online Resource Allocation Problem*. By an argument similar to that given in [11], it is clear that no online algorithm can have a bounded approximation ratio without further assumptions. In our argument, we use the following assumption which is very similar to the given constraint in [11] but defined over $q.\tau$. Assuming that $b_k = f(\tau_k.q_k)$, the requirement is that function f is a C-benevolent function. We say f is C-benevolent if we have:

1 $f(0) = 0$ and $f(p) > 0$ for all $p > 0$.

2 for $0 < \varepsilon \leq p_1 \leq p_2 \Rightarrow f(p_1) + f(p_2) \leq f(p_1 - \varepsilon) + f(p_2 + \varepsilon)$.

Under these assumptions, we will show that for arbitrary sequence of advertisers with C-benevolent profit functions, there is no online algorithm that can have a worst case ratio smaller than n. We then present an online greedy algorithm with a worst case analysis for C-benevolent profit functions, and we show that our algorithm is only a constant factor away from the best possible online algorithm.

4.2.1 Lower Bounds

In this section, we prove that there is no online algorithm with worst case ratio smaller than n for any C-benevolent profit function.

Theorem 1
There is no deterministic online algorithm for Online Resource Allocation problem *with a worst case ratio smaller than n.*

PROOF. Before describing the strategy of the adversary, we define the two types of demands(contracts) that will be used by the sequence of advertisers submitted by the adversary. For the rest of this section, we use contracts and advertisers interchangeably.

Wide: An advertiser A_i is called *wide* if $\tau_i = n$ and also $q_i = n$ meaning that they want n impressions per day for a duration of n days.

Long: An advertiser A_i is called *Long* if $\tau_i = n^2$ and $q_i = 1$.

Next, we describe the strategy of the adversary against any given online algorithm and also give an analysis to prove the lower bound of n. The strategy of the adversary against any given Heuristic H is as follows:

- Start the sequence of contracts by sending a *Wide* contract at each day.
- If H selects a *Wide* contract, then the adversary starts sending *long* contracts one per day during the active period of the *Wide* selected contract as long as it stays selected by H.
- If H selects a *Long* contract, the adversary starts sending *Wide* contracts, during the active period of the *Long* contract currently chosen by H as long as it stays active. *Wide* Contracts are submitted as soon as the previous *Wide* Contract sent by the adversary is finished.
- The adversary will stop sending any more contracts if he either sends n *Wide* contracts or n *Long* contracts or the worst case ratio is already n.

If H keeps the first *Wide* contract, then at day n the worst case ratio is already n. Now suppose H switches to the *Long* contract. Again if H keeps this contract, it will lose at day n^2 so H should switch to a *Wide* contract at some point. We can see that at the end of each round, H still only keeps one contract and also after each switch, the adversary will send at least one contract. It is clear that if H holds on to one contract and stops switching it will lose a factor of n by the finishing time of that contract. Now look at the solution after $2n$ switches. Using the given facts above and the pigeon hole principle, we can conclude that we have n contracts of the same category. Also we know that *Wide* contracts don't have conflicts with each other at all and also any set of size less than or equal to n of *Long* contracts can be scheduled together without any conflict. That means that at this point, the optimal offline solution will exceed n^3 however the solution picked by H is n^2 which gives us the desired worst case ratio for any H. □

4.2.2 The online Algorithm

In this section, we present the algorithm:

Algorithm ORA

1. If the new coming contract does not have conflict with any currently scheduled contracts, schedule it.
2. Otherwise, find[1] the set of contracts with the minimum total profit that if we dropped them from the schedule the new coming contract can be scheduled. If the profit of the new contract is more than twice the total profit of these contracts schedule it.
3. In the rest of situations, drop the new contract.

Next, we show that in the worst case, the ratio of *Online Resource Allocation Algorithm* is $8n$.

9 [1]This involves running an algorithm for a knapsack like problem on the accepted set of contracts.

Lemma 1
The Algorithm ORA *has the worst case ratio of at most* $8n + 2$ *on C-benevolent Profit functions.*

PROOF. To make the analysis more clear, we first categorize all the contracts to three groups simply based by how they are dealt with by algorithm ORA.

Final: are the contracts that are selected, and retained until the end by Algorithm *ORA*.

Dropped: are the contracts that are (initially) selected by the algorithm but dropped later on in favor of other contracts.

Ignored: are the ones that are not selected by Algorithm *ORA* at all.

We first show that the total profit of *Dropped* contracts is at most equal to the total profit of *Final* contracts. We also show that the total profit of the *Ignored* contracts belonging to the optimal solution is at most $4n$ times of the sum of *Dropped* and *Final* contracts.

Lemma 2
The total profit of the dropped *contracts is at most equal to the total profit of the* Final *contracts.*

PROOF. Consider all the *Dropped* and *Final* contracts. Create one node corresponding to each of these contracts. If a contract is dropped because of another contract, put a directed edge between them. It can be seen that the outdegree of each node corresponding to a *Dropped* contract is 1 and it is 0 for all the *Final* contracts. The structure of this graph is a directed forest rooted at *Final* contracts. We show that the total profit of all the internal nodes in each tree is at most equal to the profit of its root. The proof is by induction. Assume that for all trees of height less than h the total profit of the all the internal nodes in the subtree are less than the profit of the root. Now consider a tree of height h. We know by the way we put the edges that the profit of the root is at least twice the total profit of all the roots of the subtrees. But by induction, we also know that the total profit of the vertices of the subtrees are greater or equal to the profit of the nodes in the subtrees so that means that the profit of root of tree is greater or equal to the profit of all the nodes in the tree. Each *Dropped* contract belongs to exactly one tree since it should have exactly one outgoing edge which completes the proof. □

Next, we show that the total profit of *Ignored* contracts belonging to the optimal solution is at most $4n$ times of the total profit of *Final* and *Dropped* contracts.

Lemma 3
The total profit of Ignored *contracts in OPT are at most* $4n$ *times the total profit of* Final *and em Dropped contracts assuming the profit of each contract is computed by a C-benevolent function of the size of the contract.*

PROOF. We again use the charging method. We define the allocation for a contract and a heuristic as follows:

$Alloc(A_i, H)$**:** Assuming that we named the items available at each day by $I = i_1, \ldots, i_n$, we define $Alloc(A_i, H)$ as the the exact set of items assigned to A_i by H at each day. Without loss of generality we assume that we have 1 copy available from i_t each day.

Without loss of generality in the rest of the proof we make the following assumptions:

- We assume that for a given algorithm H and a contract A_i, the allocations can be set in a way that as long as A_i is not dropped $Alloc(A_i, H)$ does not change for the duration of each contract.
- The assigned items to each contract are staying the same during its active period. However the assigned resource might or might not be consecutive.

Consider an *Ignored* contract A_i that belongs to the optimal solution. We define the set $Conf(A_i)$ as follows:

$Conf(A_i)$**:** Consider $Alloc(A_i, OPT)$. Also consider all A_j contracts that are either *Final* or *dropped* and also $t_j \leq t_i$ and $Alloc(A_j, ORA) \bigcap Alloc(A_i, OPT) \neq \varnothing$. Call it $PConf(A_i)$. Now define $Conf(A_i) \subseteq PConf(A_i)$ as follows:

- Initialize $Conf(A_i) = PConf(A_i)$.
- Sort $A_j \in Conf(A_i)$ in the increasing order of b_j.
- If the $b_i < \sum_{A_j \in Conf(A_i)} b_j$ and $|Conf(A_i)| > 1$, remove the contract with minimum b_j from $Conf(A_i)$.
- Repeat until either $|Conf(A_i)| = 1$ or $b_i \geq \sum_{A_j \in Conf(A_i)} b_j$.

Defining the $Conf(A_i)$ as above, we can guarantee that if $|Conf(A_i) > 1|$ then $\sum_{A_j \in Conf(A_i)} b_j \leq b_i \leq 2. \sum_{A_j \in Conf(A_i)} b_j$. Next, we replace A_i with $|Conf(A_i)|$ virtual copies $A^v_{i,j}$ corresponding to each $A_j \in Conf(A_i)$ and set virtual profit $b^v_{i,j} =\leq 2.b_j$ so that $\sum_{A_j \in Conf(A_i)} b^v_{i,j} = b_i$. So $A^v_{i,j} = (t_i, \tau_i, q_i, b^v_{i,j}, 0)$.It is easy to show that it is always possible to do this. Now we can show the following lemma:

Lemma 4
Defining the price per unit of a contract(virtual contract) A_i *by* $ppu(A_i) = \frac{b_i}{\tau_i.q_i}$ *, then* $ppu(A^v_{i,j} \leq 2.ppu(A_j)$ *for all* $A_j \in Conf(A_i)$.

PROOF. If $|Conf(A_i)| = 1$ and $b_i \leq b_j$ where $A_j \in Conf(A_i)$, then since profit function is C-benevolent, we can conclude that $ppu(A^v_{i,j}) = ppu(A_i) \leq ppu(A_j)$. In the rest of the cases we can argue as follows:

Since the profit function is monotonically non decreasing, with the way we defined $Conf(A_i)$ we can conclude that $\tau_i.q_i \geq \tau_j.q_j$ $\forall A_j \in Conf(A_i)$. Now considering $A^v_{i,j}$ we know that $b^v_{i,j} \leq 2.b_j$. So we have:

$$ppu(A^v_{i,j}) = \frac{b^v_{i,j}}{\tau_i.q_i} \quad (5)$$
$$\leq \frac{2.b_j}{\tau_i.q_i} \quad (6)$$
$$\leq \frac{2.b_j}{\tau_j.q_j} \quad (7)$$
$$= 2.ppu(A_j) \quad (8)$$
$$(9)$$

□

Next, we partition $Conf(A_i)$ into two subsets:

$Tail(A_i)$: $Tail(A_i) \subseteq Conf(A_i)$ is the set of all $A_j \in Conf(A_i)$ that has overlap with $Alloc(A_i, OPT)$ at their finishing time in $Alloc(A_j, ORA)$.

$Mid(A_i)$: $Mid(A_i) \subseteq Conf(A_i)$ contains all the rest of contracts in $Conf(A_i)$.

Now the method we are using for charging is as follows:

- For each $A_j \in Conf(A_i)$ do the following:
  - If $A_j \in Tail(A_i)$ then charge A_j twice the profit of $A^v_{i,j}$ (10)
  - Otherwise, for each day t that A_j and A_i have overlap in $Alloc(A_i, OPT)$ and $Alloc(A_j, ORA)$, charge one of the common items in A_j, $2n$ times the $ppu(A^v_{i,j})$. (11)

We show that the profit of each $A^v_{i,j}$ is charged completely at least once to some $A_j \in Conf(A_i)$. Also, we will show that each A_j is not charged more than $4.n$ times considering all the *Ignored* contracts in the optimal solution.

We first show that:

Lemma 5
$\forall A_j \in Conf(A_i)$, *By the charging method described above, the total profit of* $A^v_{i,j}$ *is charged to* A_j.

PROOF. With the way we defined $A^v_{i,j}$, we know that $b^v_{i,j} \leq 2.b_j$. Now if $A_j \in Tail(A_i)$, we charge its profit twice which will directly cover $b^v_{i,j}$. Now consider the situation where $A_j \in Mid(A_i)$. Since $A_j \in Mid(A_i)$ we know that $t_i \geq t_j$ and $t_i + \tau_i \leq t_j + \tau_j$. We conclude that if $Alloc(A_i, OPT)$ and $Alloc(A_j, ORA)$ are overlapping in at least one day, they should have an overlap on every day t where $t_i \leq t \leq t_i + \tau_i$. That means in 11, we charge the profit of one unit of A_j $2n$ times for t satisfying $t_i \leq t \leq t_i + \tau_i$.Also we know that $b^v_{i,j} = ppu(A^v_{i,j}).\tau_i.q_i \leq 2.ppu(A_j).\tau_i.n$. So we can conclude that the total profit of $A^v_{i,j}$ is completely covered in our charging scheme. □

Next, we show that:

Lemma 6
Each contract that belongs to Final *or* Dropped *set, is charged at most* $4n$ *times.*

PROOF. Consider a contract A_j that belongs to *Final* or *dropped*. It can be shown that the total number of *ignored* contracts in the optimal solution that contain A_j in their $Tail(A_i)$ are at most n. The reason is that A_j can occupy at most n resources on its finishing day and since we are considering a feasible fixed allocation of the optimal solution, each of these resource can belong to at most one contract in the optimal solution and in total at most n contracts in OPT have overlap with A_j at its finishing time. Also, each resource of A_j will be charged once and for an amount equal to $2n$ times the price per unit of that resource because the charged resource should belong to the overlap of A_j and A_i and A_i belongs to the optimal solution and we consider a fixed feasible allocation of OPT, so no other A_k where $k \neq i$, can have overlap with A_j at the same resource. Therefore in 11, each unit will be charged at most $2n$ times. Putting all these together, we can conclude that each A_j in either *Final* or *Dropped* is charged at most $4n$ times. □

The final goal is to compare the profit of *Final* contracts with the total profit of optimal. Partition the total profit of OPT to $OPT_f + OPT_d + OPT_i$ based on the category that they belong to in ORA. Also call the profit of *Final* contracts P_f and profit of *dropped* contracts, P_d. We know that $OPT_f \leq P_f$, $OPT_d \leq P_d \leq P_f$ and finally $P_i \leq 4n.(P_f + P_d) \leq 8n.P_f$. So $OPT_f + OPT_d + OPT_i \leq (8n+2)P_f$ which completes the proof.

4.3 Simulation

In this subsection we test the performance of our algorithm on real data derived from the Yahoo! display advertisement business. To do so, we selected four different types of impressions and considered all those contracts they could be used to satisfy. The types of impressions represent different sets of *properties* (e.g. mail, Finance, etc.) and *positions* (e.g. top, bottom, side). We then partitioned the business period into rounds. This partition determined the value of T for each dataset. Data is derived from proprietary but real contract data from 2007.

The simulation methodology is the following. We first assumed that all advertisers were interested *only* in being delivered the minimum number of impressions. Hence $V_k = 0$ for all advertisers. Next, we parameterized the simulation by n, the number of impressions available per round, and ran the simulation for 5 distinct values of n. In order to calculate the optimal revenue, we attempted to solve the following integer program:

$$\max_x \sum_{k=1}^{K} b_k x_k \text{ subject to} \quad (12)$$

$$\sum_{k=1}^{K} q_k a_{t,k} x_k \leq n, \text{ for all } 1 \leq t \leq T \quad (13)$$

$$x \in \{0,1\}^K \quad (14)$$

where $a_{t,k} = 1$ if advertiser k is interested in impressions during round t, and $a_{t,k} = 0$ otherwise. The variable $x_k = 1$ indicates that a contract will be sold to advertised A_k. Note that the number of variables is equal to K, and that the number of constraints is given by T, the number of periods.

Since the number of contracts may be large, it is important to note that the ability to solve the previous integer program depends both on the number of constraints (as given by Inequalities 13) active at optimality, and on the total number of contracts selected at an optimal allocation. We capture both effects by varying the number of impressions available per round. If n is large enough, all advertisers can book a contract. As n decreases, the number of rounds where a potential contention exists increases, and the solution becomes "lumpier".

In the course of solving the integer program, we also obtained the solution to the LP relaxation, where the Inequalities 14 are replaced with

$$0 \leq x_k \leq 1, \text{ for all } 1 \leq k \leq K \quad (15)$$

Since for some of the datasets the integer program produced a *worse solution* than that of our algorithm, we report the results of our algorithm as compared to the solution obtained by the LP relaxation. We note that we used default settings for the IP solver, which might account for its inability to handle some of the datasets. The results of our simulations are summarized in Table1 1-4.

Table 1: Results for the first dataset

n	Number Booked	Performance loss
n_1	2450	0
n_2	1709	3.71
n_3	803	14.2
n_4	88	10.5
n_5	33	8.36

The parameters for each dataset are summarized in Table 5.

Table 2: Results for the second dataset

n	Number Booked	Performance loss
n_1	1581	0
n_2	1070	10.4
n_3	551	12.6
n_4	169	14.4
n_5	111	37.9

Table 3: Results for the third dataset

n	Number Booked	Performance loss
n_1	9169	0
n_2	9124	0.05
n_3	8323	0.52
n_4	6615	0.84
n_5	1285	4.99

The performance loss is given in percentage points with respect to the LP solution. Note that our algorithm performs orders of magnitude better than the guaranteed performance (note that even in the case on $n = 1$, the performance guarantee would be of 25%, or a performance loss of 75%!). This indicates that the instances typically encountered in real setting are far from those necessary to make our algorithm perform poorly.

In Table 6 we compare the performance of the IP solution to that of the LP solution. Note that for datasets two to four, the performance of the integer programming solution was sometimes below that of our solution, and in some instances was not even available. We used the open source branch and cut code Cbc from the COIN-OR repository [9], with default parameter settings, but with the attempted addition of the probing, Gomory, knapsack, redsplit and clique cuts. In general, the number of cuts obtained was small. The branch and bound algorithm was run until either the 10th integer solution was obtained, or twenty minutes have elapsed.

Table 4: Results for the fourth dataset

n	Number Booked	Performance loss
n_1	8826	0
n_2	6355	20.0
n_3	4603	23.6
n_4	850	8.09
n_5	436	6.72

Table 5: Parameters of the datasets

Parameters	Dataset1	Dataset2	Dataset3	Dataset4
n_1	5M	5M	1B	500M
n_2	1M	1M	500M	100M
n_3	500k	500k	100M	50M
n_4	100k	100k	50M	10M
n_5	50k	50k	10M	5M
# contracts	2450	1581	9169	8826
T	12	90	90	90

5. REFERENCES

[1] A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *Proceedings of ACM Symposium on the Theory of Computing*, pages 114–121, 2007.

[2] N. Bansal and M. Sviridenko. The santa claus problem. In *Proceedings of ACM Symposium on the Theory of Computing*, pages 31–40, 2006.

[3] G. Calinescu, A. Chakrabarti, H. J. Karloff, and Y. Rabani. Improved approximation algorithms for resource allocation. In *Proceedings of the 9th International Integer Programming and Combinatorial Optimization Conference, IPCO 2002*, pages 401–414, 2002.

[4] L. Epstein and A. Levin. Improved randomized results for the interval selection problem. In *ESA*, 2008.

[5] U. Feige, N. Immorlica, V. S. Mirrokni, and H. Nazerzadeh. A combinatorial allocation mechanism with penalties for banner advertising. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008*, pages 169–178, 2008.

[6] S. Fung, C. Poon, and F. Zheng. Improved randomized online scheduling of unit length intervals and jobs. In *WAOA*, 2008.

[7] J. James D. Dana. Advance-purchase discounts and price discrimination in competitive markets. *Journal of Political Economy*, 106(21):395–422, 1998.

[8] R. Lewis and D. Reiley. Retail advertising works! measuring the effects of advertising on sales via a controlled experiment on yahoo! Technical report, Yahoo! Research, 2008.

[9] COIN-OR Foundation: `http://www.coin or.org`.

[10] V. Srinivasan, C. S. Park, and D. R. Chang. An approach to the measurement, analysis and predition of band equity and its sources. *Management Science*, 51(9):1433–1448, 2005.

[11] G. J. Woeinger. On-line scheduling of jobs with fixed start and end times. *Theoretical Computer Science*, 130(1):5–16, 1994.

APPENDIX

A. MULTIPLE IMPRESSION TYPES

In this appendix we present the relevant utility model when more than one impression type is available. This is the model used in Subsection 3.1. Assume there are $L > 1$ impression types, corresponding to different advertising selection criteria used by advertisers. For instance, an impression type could represent the set of all finance related webpages, and another one the set of all sports related webpages. For ease of exposition, we assume each impression has a unique type (note that, in general, a given impression could potentially be of several types, specially in the presence of behavioral targeting).

We note by n_t^ℓ the number of impressions of type $\ell \in \{1, \ldots, L\}$ available at round $t \in \{1, \ldots, T\}$. Thus the set of possible allocations is now given by

$$\mathbf{S} = \prod_{\ell=1}^{L} \left[\prod_{t=1}^{T} \{0, \ldots, n_t^\ell\} \right]$$

where $s \in \mathbf{S}$ represents a possible allocation of impressions to a given advertiser with $s(\ell, t) \in \{0, \ldots, n_t^\ell\}$ represents the number of impressions of type ℓ allocated at round t to the advertiser. We note by $s(\ell) \in \prod_{t=1}^{T} \{0, \ldots, n_t^\ell\}$ the set of impressions of type ℓ allocated in s. Let $N_\ell = \sum_{t=1}^{T} n_t^\ell$ be the total supply over the business period of impressions of type ℓ.

Again, we index advertisers by $k \in \{1, \ldots, K\}$ and note them A_k. An *allocation* $\mathcal{S} = \{s_1, \ldots, s_K\}$ is a k-partition of the set impressions such that,

Table 6: Performance of the Integer Program solution

n	Dataset1	Dataset2	Dataset3	Dataset4
n_1	0	0	0	0
n_2	0.04	5.01	0.02	N/A
n_3	0.10	8.53	17.7	N/A
n_4	1.77	36.7	8.27	91.1
n_5	2.80	32.1	9.54	N/A

- for all $k \in \{1, \ldots, K\}$, $s_k \in \mathbf{S}$; and
- for all $\ell \in \{1, \ldots, L\}$, $\sum_k s_k(\ell) \leq (n_1^\ell, \ldots, n_T^\ell)$.

The vector s_k represents the set of impressions allocated to advertiser A_k, with $s_k(\ell)$ representing the set of impressions of type ℓ. The second condition states that, for any type ℓ, we do not allocate more impressions of that type during a given round than available.

In order to meet branding requirements, we assume that advertiser A_k is interested only in a subset $L_k \subseteq \{1, \ldots, L\}$ of the impressions' types available. More precisely, starting from round t_k and for τ_k consecutive rounds, advertiser A_k is interested in receiving at least a given number δ_k of impressions per round of any type $\ell \in L_k$. The value drawn from receiving those impressions is b_k.

If A_k meets its branding constraints and receives an additional q_ℓ impressions of type $\ell \in \{1, \ldots, L\}$, then A_k draws an additional $V_k^\ell(q_\ell)$ units of utility. More formally, advertiser A_k is fully characterized by the $5 + L$-tuple $(L_k, t_k, \tau_k, \delta_k, b_k, V_1, \ldots, V_L)$ where

- $L_k \in \{1, \ldots, L\}$ is the set of impressions types' advertiser A_k is interested in for branding purposes; and
- $t_k \in \{1, \ldots, T\}$ is the *start date* advertiser A_k is interested in; and
- $\tau_k \in \{1, \ldots, T\}$ is the *duration* advertiser A_k is interested in; and
- $\delta_k \in \{1, \ldots, n\}$ is the *minimum impressions per round* advertiser A_k seeks to receive starting at t_k and for τ_k rounds from the types specified in L_k; and
- $b_k \in \mathbb{R}^+$ is the *utility* drawn from receiving δ_k impressions per round during τ_k consecutive rounds; and
- for all $\ell \in \{1, dots, L\}$, V_k^ℓ is the *additional value* function for impressions of type ℓ

  $$V_k^{ell} : \{0, 1, \ldots, N_\ell\} \to \mathbb{R}$$
  $$q \mapsto V_k^\ell(q)$$

  where $V_k^\ell(q)$ is the value drawn by advertiser A_k of being allocated q additional impressions of type ℓ *given* that it is guaranteed to receive δ_k impressions per round during τ_k consecutive rounds of type from the set L_k. Note that $V_k^\ell(q) < 0$ can model ad-fatigue.

Given an allocation $s_k \in \mathbf{S}$, if for all $t \in [t_k, t_k + \tau_k - 1]$, it holds that

$$\sum_{\ell \in L_k} s(\ell, t) \geq \delta_k, \tag{16}$$

then the utility to agent A_k is

$$U(s_k; A_k) = b_k + \sum_{\ell=1}^{L} V_k^\ell \left(\sum_{t=1}^{T} s'_k(\ell, t) \right) \tag{17}$$

where s'_k is the residual allocation obtained by subtracting the impressions used for branding purposes to s_k. Note that there are potentially several ways to define s'_k. We assume s'_k is defined such that the utility to A_k is maximized.

Otherwise, if any of the Inequalities 16 is not satisfied, the utility to agent A_k is assumed to be nil.

Again, the previous utility function simply says that, if agent A_k is *not* given an allocation where it receives at least δ_k impressions of types in L_k per round during rounds in $[t_k, t_k + \tau_k - 1]$, then A_k's utility is nil. This accounts for branding activities linked to display advertising.

Notice that the utility drawn from impressions of different types not used to satisfy branding constraints are added together. This is consistent with the idea that, once branding constraints are satisfied, advertisers treat impressions as substitutes (not necessarily perfect substitutes).