

A Combinatorial Allocation Mechanism With Penalties For Banner Advertising

Uriel Feige
Weizmann Institute
Rehovot 76100, Israel
uriel.feige@weizmann.ac.il

Nicole Immorlica
Centrum voor Wiskunde en
Informatica
Amsterdam, the Netherlands
immorlic@cwi.nl

Vahab S. Mirrokni
Microsoft Research
Redmond, WA 98052
mirrokni@microsoft.com

Hamid Nazerzadeh
Stanford University
Stanford, CA 94305
hamidnz@stanford.edu

ABSTRACT

Most current banner advertising is sold through negotiation thereby incurring large transaction costs and possibly sub-optimal allocations. We propose a new automated system for selling banner advertising. In this system, each advertiser specifies a collection of host webpages which are relevant to his product, a desired total quantity of impressions on these pages, and a maximum per-impression price. The system selects a subset of advertisers as *winners* and maps each winner to a set of impressions on pages within his desired collection. The distinguishing feature of our system as opposed to current combinatorial allocation mechanisms is that, mimicking the current negotiation system, we guarantee that winners receive at least as many advertising opportunities as they requested or else receive ample compensation in the form of a monetary payment by the host. Such guarantees are essential in markets like banner advertising where a major goal of the advertising campaign is developing brand recognition.

As we show, the problem of selecting a feasible subset of advertisers with maximum total value is inapproximable. We thus present two greedy heuristics and discuss theoretical techniques to measure their performances. Our first algorithm iteratively selects advertisers and corresponding sets of impressions which contribute maximum marginal per-impression profit to the current solution. We prove a *bi-criteria approximation* for this algorithm, showing that it generates approximately as much value as the optimum algorithm on a slightly harder problem. However, this algorithm might perform poorly on instances in which the value of the optimum solution is quite large, a clearly undesirable failure mode. Hence, we present an adaptive greedy algorithm which again iteratively selects advertisers with maximum marginal per-impression profit, but additionally reassigns impressions at each iteration. For this algorithm, we prove a *structural approximation* result, a newly defined framework for evaluating heuristics [10]. We thereby prove that this algorithm has a better performance guarantee than the simple greedy algorithm.

Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity; J.4 [Computer Applications]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithm, Theory, Economics

Keywords

Internet Advertising, Supply Guarantee, Combinatorial Auctions, Structural Approximation

1. INTRODUCTION

Online advertising is one of the most profitable business models for Internet services to date, accounting for annual revenues approaching \$17 billion dollars according to the Internet Advertising Bureau [21]. According to the same report, 22% of this revenue comes from banner advertising, or graphic-based advertisements which appear embedded in content pages on a host website. Current models of banner advertising require advertisers to negotiate rates directly with the sales representatives of the host on a monthly basis. These negotiations involve several parameters. First, the advertiser specifies a subset of content pages which are relevant to his or her ad. Next, the advertiser requests a desired number of impressions, or advertising opportunities, on pages from among his or her specified subset, and a per-impression price. Finally, if the negotiation is successful and the contract is accepted, the advertisement is shown on some subset of the specified content pages. By accepting a contract the supplier is committed to the supply. If the supply is not met, the bidder is compensated in two ways: he is not charged for the parts that are not met, and moreover, he gets them for free in the next time period.

This process requires a lot of overhead for advertisers who must negotiate with sales representatives, and a lot of guesswork for sales representatives who must decide based on intuition which advertising contracts to accept. As a result, small advertisers tend to be locked out of the system, unable to bear the cost of the smallest-sized contracts, and the system overall suffers additional inefficiencies from the sub-

optimal decisions of the sales representatives. In this paper, we consider automating this system. We design a system which each month considers the supply of page views and a set of bids, each specifying a desired set of pages, a desired number of impressions, and a per-impression value, and then selects a subset of bids which maximizes the value while sustaining penalties for under-allocated impressions. Whereas in practice, the penalties are realized in the form of free impressions, in our system we represent these penalties with a monetary payment from the host to the advertiser equal to a specified multiple of the per-impression value for each under-allocated impression. The penalty factor reflects an estimate of how much next month’s impressions are going to be worth.

There are several factors which make this system harder to design than general combinatorial allocation mechanisms. The most significant difference comes from the fact that the host must “guarantee” supply by paying a penalty for under-allocated impressions. Such guarantees are quite characteristic of the banner advertising market, and are important in branding campaigns where it is essential for the advertiser to receive at least some minimum number of impressions – the McDonald’s arches or the Nike swoosh are recognizable precisely because they are ubiquitous. However, these guarantees also make the underlying optimization problem hard to approximate: for example, if the penalty is very large, then each winner must receive all of his requested impressions and so we can reduce our problem from the well-known set packing problem [15]. In fact, as we show, even for small penalties, our problem is not approximable within any constant factor.

We therefore focus on designing heuristics with varying sorts of provable guarantees. A common approach to combat inapproximability is to develop *bi-criteria* approximations, or algorithms that do almost as well as the optimum solution on a slightly harder problem. For example, in β -balanced graph partitioning, the goal is to cut the minimum number of edges to divide a graph on n vertices into two parts, each with at most βn vertices. The seminal paper of Leighton and Rao [18] provides a bi-criteria approximation algorithm for $(2/3)$ -balanced graph partitioning which cuts at most a factor of $O(\log n)$ more edges than the optimum solution on the harder problem of bisection, or $(1/2)$ -balanced graph partitioning. This was improved to $O(\sqrt{\log n})$ by Arora et al. [3]. The bi-criteria framework has also been applied to min sum clustering [5] and selfish routing [22].

In our case, we apply the bi-criteria framework by designing an algorithm that does approximately as well as the optimum algorithm with larger penalties. Our algorithm is extremely simple and trivial to implement: it selects at each iteration a new winner and a corresponding set of impressions with maximum marginal per-impression profit. We show that this algorithm gets at least 23% of the optimum profit on a problem with 2.26 times the penalty (i.e., whereas our algorithm just pays an advertiser his per-impression bid for each under-allocated impression, the optimum must pay 2.26 times the per-impression bid).

Despite the bi-criteria guarantee, we note that this greedy algorithm may perform poorly when the optimum solution has large value. These are exactly the instances in which we would like to perform well. Hence, we present a refined greedy algorithm which involves solving a flow computation and prove guarantees on its performance in the *structural*

approximation [10] framework, a newly developed framework for evaluating the approximation factor of heuristic algorithms. This framework allows one to evaluate the approximation factor of an algorithm based on the structure of the optimal (or any) solution, and not only the value of the optimal solution. In particular, it provides a performance guarantee that improves as the performance of the optimum solution improves, and unlike the bi-criteria framework, it provides these guarantees with respect to the optimum *on the same problem instance*.

Our refined greedy algorithm iteratively selects advertisers with maximum marginal per-impression profit, as before, but additionally considers re-allocating impressions to already selected advertisers at each step. In evaluating this algorithm, we consider the fraction of demand that the optimum satisfies for each advertiser. We show as this fraction increases, hence increasing the optimum value, our algorithm’s value also increases. For example, if the optimum always satisfies 75% of the demand of each advertiser, then our algorithm gets at least 10% of the optimum, whereas if the optimum satisfies 99% of the demand of each advertiser, then our algorithm gets at least 30% of the value of the optimum. See Section 5.1 for the exact trade-off curve.

We note that the refined greedy algorithm outperforms the simple greedy algorithm on many classes of instances. However, this improved performance comes at the expense of increased difficulty in the implementation (and an additional linear factor in the running time).

It is important to note that we ignore incentives throughout this work, i.e., we analyze the value of our allocation mechanism in an idealized world where advertisers truthfully report their types and not in an economic equilibrium. While incentive issues are very important in practice, they tend to have less effect in highly competitive settings. In successful banner advertising systems, where lots of advertisers compete for limited supply, we intuitively expect incentives to play a minimal role. We leave it as further work to study incentives in our proposed systems and/or calculate payments to make our proposed systems truthful.

1.1 Related Work

The problem of maximizing welfare is a central problem in algorithmic mechanism design. This problem, even for special cases such as single-minded bidders, is known to be hard even to approximate [23, 17, 16, 7, 8]. However, under some conditions such as submodularity or subadditivity of valuations, constant-factor approximation algorithms have been developed [8, 9]. For several problems with packing constraints linear programming-based and greedy constant-factor approximation algorithms have been proposed [1, 13].

In the context of Internet advertising, sponsored search auctions have been studied extensively. The closest works to ours in this literature are the $1 - \frac{1}{e}$ -competitive algorithms that have been designed for maximizing revenue for online allocation of advertisement space [20, 6, 19].

Our work is distinguished from previous work in the literature by the existence of supply guarantees, implemented through penalties. These penalties reflect the non-linearity of bidder valuations, a characteristic of advertisers in banner advertising markets.

1.2 Organization

This paper is organized as follows. We give a formal def-

inition of the model in Section 2. In this section, we also give the following basic observation about our problem: in the guaranteed banner advertisement problem, if the set of served advertisers is fixed, the optimal allocation can be found in polynomial time. This indicates that the challenging part of the problem is to find the set of advertisers that should be served to maximize the value. In Section 3, we prove that the problem does not admit a constant-factor approximation algorithm even for weak supply guarantee conditions. Next, in Section 4, we give a bi-criteria approximation algorithm for the problem. In Section 5, we give an adaptive greedy algorithm and analyze it in the structural approximation framework.

2. MODEL

The banner advertisement problem can be modeled as follows. We are given a set \mathcal{A} of m advertisers $\{1, \dots, m\}$ and a set \mathcal{U} of advertising opportunities that can be allocated to them. Each advertising opportunity is an impression of an advertisement on a webpage. For the rest of the paper, we refer to each advertising opportunity as an *item*. Without loss of generality, we assume all items of \mathcal{U} are distinct. The bid of an advertiser i is a tuple (I_i, d_i, b_i) where $I_i \subseteq \mathcal{U}$ specifies the subset of items in which advertiser i is interested¹, d_i specifies the number of items the advertiser requires, and b_i specifies the per-unit price the advertiser is willing to pay (up to his required number of units). Allocating more than d_i items to advertiser i has no marginal benefit.

Following the convention of current banner advertising systems, we require our system to *guarantee* the supply of an advertiser. That is, by accepting the bid of advertiser i , we enter a contract for the sale of d_i items from the set I_i . Should we fail to fulfill this contract, we are required to pay a *penalty*. This penalty can be a fixed price paid for every under-allocated impressions; or more commonly, can be filled by delivering the same number of impressions for free during the following period. Both of these penalty schemes can be formalized in the following way: If the advertiser requested d_i units and was allocated only x_i units for some $x_i < d_i$, then the system collects only

$$b_i x_i - \lambda b_i (d_i - x_i) = ((\lambda + 1)x_i - \lambda d_i) b_i \quad (1)$$

in value, where λ is a fixed constant. The special case of $\lambda = 1$ can be interpreted as delivering the same number of impression for free the following month. Henceforth, we will be primarily concerned with the case $\lambda = 1$, unless otherwise stated, although all results can be appropriately generalized to other values of λ .

Now we can formalize the banner advertisement problem in the following way: Given a set of advertisers \mathcal{A} and their bids, $\{(I_i, d_i, b_i)\}$, find a subset of advertisers, or winners, for which accepting their bids maximizes the value. This problem is captured by the following mathematical program. Let y_i be the indicator variable that advertiser i is a winner; z_{ij} be the indicator variable that item j is allocated to advertiser i ; and x_i be the total number of items allocated to advertiser i .

¹Each advertiser is interested in advertising on webpages with specific content such as finance, sports, etc. Advertising can be even more targeted by using demographic information of the current viewer of the webpage such as gender, geographic location, or age.

$$\begin{aligned} & \text{maximize} && \sum_{i \in \mathcal{A}} y_i (2x_i - d_i) b_i && (2) \\ & \text{subject to} && \forall j \in \mathcal{U} : \sum_{i \in \mathcal{A}} z_{ij} \leq 1 \\ & && \forall i \in \mathcal{A} : x_i = \sum_{j \in I_i} z_{ij} \\ & && \forall i \in \mathcal{A} : x_i \leq d_i \\ & && \forall i \in \mathcal{A}, \forall j \in \mathcal{U} : z_{ij} \in \{0, 1\} \\ & && \forall i \in \mathcal{A} : y_i \in \{0, 1\} \end{aligned}$$

As we show in the next section, it is NP-hard to even approximate the solution of this optimization program within any constant ratio. The main difficulty comes from the multiplicative terms $y_i x_i$ in the objective function of the program above. However, we show that once the set of winning advertisers is determined (i.e., once we fix the $\{y_i\}$ variables), the resulting program can be solved combinatorially using a maximum weighted matching algorithm.

LEMMA 2.1. *Given the set T of winning advertisers, the problem of finding the optimum allocation of items to advertisers in T can be optimally solved in polynomial time.*

PROOF. We can rewrite the above mathematical program by setting $y_i = 1$ for $i \in T$ and $y_i = 0$ otherwise, and by setting $x_i = \sum_{j \in I_i} z_{ij}$. The negative term in the resulting objective function is now a constant, and so the problem reduces to the following program:

$$\begin{aligned} & \text{maximize} && \sum_{i \in T} \sum_{j \in I_i} 2z_{ij} b_i \\ & \text{subject to} && \forall j \in \mathcal{U} : \sum_{i \in T} z_{ij} \leq 1 \\ & && \forall i \in T : \sum_{j \in I_i} z_{ij} \leq d_i \\ & && \forall i \in T, \forall j \in \mathcal{U} : z_{ij} \in \{0, 1\} \end{aligned} \quad (3)$$

This integer program is equivalent to a maximum weighted b -matching problem, or f -factor problem, in which, given a weighted graph G , the goal is to find a maximum-weight subgraph of G in which the degree of each node i is at most $f(i)$. The mapping is as follows: consider a bipartite graph $G(V, E)$ with node set $V(G) = \mathcal{U} \cup T$, edge set $E(G) = \{(i, j) | i \in T, j \in I_i\}$, and edge weights $w(i, j) = 2b_i$. The solution to the integer program is then the maximum weighted f -matching in which $f(i) = d_i$ for $i \in T$ and $f(j) = 1$ for $j \in \mathcal{U}$. We can therefore use the combinatorial or LP-based algorithms for f -matching (see, e.g., [25, 24, 4]) to solve our problem. \square

The above discussion shows that the challenging part of the banner advertising problem is to decide the set T of winning advertisers.

3. HARDNESS OF APPROXIMATION

In this section, we show that the banner advertising problem is NP-hard to approximate within any constant ratio. In other words, for any constant $0 < c \leq 1$, it is not possible

to find an allocation with value at least c times the optimum value (defined as the value of the mathematical program (2)) in polynomial time unless $P = NP$.

This result is immediate for arbitrarily large penalties via a reduction from the inapproximable *maximum set-packing problem* [15], i.e., the problem of finding the maximum number of disjoint sets in a given set family. For arbitrary penalties, we provide a reduction from the *k -uniform regular set cover problem*. In this problem, we are given a family of sets over a universe of elements. Each set in the family has exactly k elements, and each element in the universe is covered by exactly k sets. The goal is to select the minimum number of sets which cover all the elements. It is shown in Feige et al. [14], Theorem 12, that:

PROPOSITION 3.1. [11, 14] *For every choice of constants $s_0 > 0$ and $\varepsilon > 0$, there exists k dependent on ε and instances of k -uniform regular set cover with n elements on which it is NP-hard to distinguish between the case in which all elements can be covered by $t = \frac{n}{k}$ disjoint sets, and the case in which every $s \leq s_0 t$ sets cover at most a fraction of $1 - (1 - \frac{1}{t})^s + \varepsilon$ of the elements.*

By introducing an advertiser with bid $(I_i, |I_i|, 1)$ for each set I_i in the set cover instance, there is a trivial reduction from the problem above which shows that it is NP-hard to compute the optimum value of the banner advertising problem. In the case in which all elements can be covered by $t = \frac{n}{k}$ disjoint sets, the optimum value would be n , but, the case in which every $s \leq s_0 t$ sets cover at most a fraction of $1 - (1 - \frac{1}{t})^s + \varepsilon$ of the elements, the optimum value would be less than n .

In order to prove hardness of approximation, we find an upper bound on the value of the optimum solution in the second case, which leads to the theorem below.

THEOREM 3.2. *The banner advertising problem is NP-hard to approximate within any constant ratio $0 < c \leq 1$.*

The theorem is proved by showing that if one can approximate the solution within a constant ratio, then it is possible to distinguish between the instances of the set cover problem described above. The complete proof is given in appendix A. Although the proof in the appendix is presented assuming the penalty λ is equal to one, the result holds for any constant linear penalty.

4. A GREEDY ALGORITHM

In lieu of the above hardness result, we present a greedy heuristic which at each step assigns items to an advertiser that maximizes the profit-per-item. The description of the algorithm is given in Figure 1. Note that it is important to choose an advertiser that maximizes profit-per-item. The example below shows that a naive greedy algorithm which chooses the advertiser with maximum overall profit performs quite poorly even in “easy” instances.

Example Suppose there are $n + 1$ advertisers and n items. Advertiser i , $1 \leq i \leq n$, is interested only in item i and bids 1 for it. Advertiser $n + 1$ bids $1 + \varepsilon$ for receiving all the n items. While the optimal solution yields a value of n by allocation items 1 to n to advertisers 1 to n , the naive greedy algorithm obtains a value of just $1 + \varepsilon$.

The value that algorithm GREEDY obtains from the instance in the previous example is equal to the optimum value.

4.1 Performance Guarantee: Bi-Criteria Approximation

We analyze algorithm GREEDY by developing *bi-criteria approximations* for our problem. To do so, we compare the value of our solution to the value of the optimal solution with a greater penalty. We say an algorithm is an (α, β) -approximation if it always outputs a solution with value at least β times that of the optimal solution with (larger) penalties α . More formally, an algorithm is an (α, β) -approximation for the problem of selling banner ads if it can always obtain a β fraction of the value of the linear program below – this is the same as LP 2 but with a different objective.

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathcal{A}} y_i ((\alpha + 1)x_i - \alpha b_i d_i) b_i & (4) \\ \text{subject to} \quad & \forall j \in \mathcal{U} : \sum_{i \in \mathcal{A}} z_{ij} \leq 1 \\ & \forall i \in \mathcal{A} : x_i = \sum_{j \in I_i} z_{ij} \\ & \forall i \in \mathcal{A} : x_i \leq d_i \\ & \forall i \in \mathcal{A}, \forall j \in \mathcal{U} : z_{ij} \in \{0, 1\} \\ & \forall i \in \mathcal{A} : y_i \in \{0, 1\} \end{aligned}$$

In other words, while our algorithm pays a penalty of 1 for under-allocation, the optimal solution is charged a penalty of α for some $\alpha > 1$.

For algorithm GREEDY we can prove the following theorem:

THEOREM 4.1. *For $\alpha \geq \frac{\ln 2}{1 - \ln 2} \approx 2.26$, the approximation ratio of GREEDY algorithm is $\frac{1 - \ln 2}{2 - \ln 2} \approx 0.23$. For $1 < \alpha < \frac{\ln 2}{1 - \ln 2}$, the approximation ratio is*

$$\frac{2x - 1 - \ln 2x}{(\alpha + 3)x - \alpha - 1 - \ln 2x} \quad (5)$$

where $x \in [\frac{\alpha}{\alpha + 1}, 1]$ is the root of the equation $1/x + (1 + 1/\alpha) \ln 2x = 2$.

PROOF. For any $\alpha > 1$, let OPT_α be an optimal solution for the penalty α . We provide lower bounds on the approximation ratio of GREEDY and OPT_α . Let \mathcal{A} be the set of all advertisers who contribute a positive profit to OPT_α . For $i \in \mathcal{A}$, let S_i be the set OPT_α allocated to advertiser i . To simplify notation, we use s_i to denote $\frac{|S_i|}{d_i}$. Without loss of generality, we may assume $\frac{\alpha}{\alpha + 1} < s_i \leq 1$.

In the analysis of GREEDY algorithm, we will partition each item into two subitems: the P subitem that takes up a p fraction of the item, and the Q subitem that takes up a $(1 - p)$ fraction of the item; we will determine the value of p later. We will develop lower bounds on the profit of GREEDY by a case analysis. In each case, we will compare the profit OPT_α gets to one of two kinds of profit that GREEDY gets: only to the part computed over P subitems, or only to the part computed over Q subitems. Summing over all

GREEDY Algorithm:

Initialize the set of allocated items Q to be the empty set

While there exists an unassigned advertiser:

1. For each unassigned advertiser i :
 - (a) Let $J'_i = I_i \setminus Q$ be the set of remaining items that we can allocate to i and $J_i \subseteq J'_i$ be an arbitrary subset of size $\min(d_i, |J'_i|)$.
 - (b) Let $p_i = (2 - d_i/|J_i|)b_i$ be the profit-per-item obtained by allocating J_i to i .
2. Choose the advertiser i^* whose per-item profit p_{i^*} is maximum.
3. If p_{i^*} is positive, then set i^* to be a winner, allocate J_{i^*} to i^* , and set Q to be $Q \cup J_{i^*}$; otherwise terminate the algorithm.

Figure 1: GREEDY algorithm

advertisers, the P part and Q part of each item, and hence the whole item, is counted at most once.

For $i \in \mathcal{A}$, there are two cases:

1. GREEDY allocates strictly less than $|S_i| - \lceil d_i/2 \rceil$ items from S_i . In this case we consider the profit GREEDY obtains from P subitems assigned to i .

When j elements of S_i are remaining, GREEDY could allocate these elements to i with profit-per-item $\frac{2j-d_i}{j}b_i$. This profit is positive for $\lceil d_i/2 \rceil < j \leq |S_i|$. Therefore, in this case, GREEDY must have allocated a nonempty set to advertiser i since otherwise GREEDY could get positive per-item profit by adding i to the solution and allocating the unallocated items of S_i to i . Furthermore, as GREEDY has items of S_i available when assigning items to i , it follows that the demand d_i of advertiser i is fully satisfied. Hence GREEDY gets complete profit for P subitems, giving a profit of $pd_i b_i$.

2. GREEDY allocates at least $|S_i| - \lceil d_i/2 \rceil$ items from S_i . In this case we consider the profit GREEDY gets from Q subitems of S_i .

Note, GREEDY must have obtained a profit of at least $\frac{2j-d_i}{j}b_i$ from the $(|S_i| - j + 1)$ 'th item allocated from S_i as this is the per-item profit GREEDY obtains by allocating the remainder of S_i to i . Summing over j , $\lceil d_i/2 \rceil + 1 \leq j \leq |S_i|$, GREEDY must have allocated these items with total profit at least:

$$\begin{aligned} \sum_{j=\lceil d_i/2 \rceil+1}^{|S_i|} \frac{2j-d_i}{j} b_i &\geq \int_{d_i/2}^{|S_i|} \frac{2x-d_i}{x} b_i dx & (6) \\ &= \int_{d_i/2}^{|S_i|} \left(2 - \frac{d_i}{x}\right) b_i dx \\ &= (2|S_i| - d_i - d_i(\ln |S_i| - \ln \frac{d_i}{2})) b_i \\ &= (2s_i - 1 - \ln 2s_i) d_i b_i \end{aligned}$$

Taking into account that this profit comes only from the Q part of items, this gives a profit of $(1-p)(2s_i - 1 - \ln 2s_i) d_i b_i$.

For each advertiser i , exactly one of the above cases happens. Furthermore, for each item, the P part is counted at

most once since the assignment in case 1 is a subset of the solution for GREEDY and hence disjoint. Similarly, the Q part is also counted at most once since the S_i considered in case 2 is a subset of the optimum solution and hence disjoint. Therefore, we have shown that GREEDY gets a profit of at least:

$$\sum_{i \in \mathcal{A}} \min\{p, (1-p)(2s_i - 1 - \ln 2s_i)\} d_i b_i. \quad (7)$$

The profit of OPT_α from S_i is $((\alpha+1)s_i - \alpha)d_i b_i$. Therefore, the ratio of the profit of GREEDY to OPT_α is at least $\min\{p, (1-p)\gamma\}$, where γ is defined as

$$\gamma = \min_i \left\{ \frac{2s_i - 1 - \ln 2s_i}{(\alpha+1)s_i - \alpha} \right\} \quad (8)$$

Choosing $p = \frac{\gamma}{1+\gamma}$ gives the approximation ratio $\frac{\gamma}{1+\gamma}$.

We find a lower bound on γ using the first and second order conditions. The details of calculations are given in appendix B. For $\alpha \geq \frac{\ln 2}{1-\ln 2}$, γ takes its minimum for $s_i = 1$. Plugging in (8), we get $\gamma = \frac{2-1-\ln 2}{(\alpha+1)s_i-\alpha} = 1 - \ln 2$ and $p = \frac{1-\ln 2}{2-\ln 2}$

For $1 < \alpha < \frac{\ln 2}{1-\ln 2}$, as we prove in appendix B, there exists a point x in $[\frac{\alpha}{\alpha+1}, 1]$ for which $s_i = x$ gives the minimum value of γ . This point is the root of the equation below which is derived by taking the derivatives of (8):

$$\alpha/x - 2\alpha + (\alpha+1)\ln 2x = 0.$$

Plugging this value for γ into $\frac{\gamma}{1+\gamma}$, we get (5) as a lower bound on the approximation ratio for $1 < \alpha < \frac{\ln 2}{1-\ln 2}$. \square

We give an example which shows the analysis is tight for $\alpha \geq \frac{\ln 2}{1-\ln 2}$. In this example, the optimal solution allocates all of the items to the advertisers without paying any penalty. However, the GREEDY algorithm obtains just a $\frac{1-\ln 2}{2-\ln 2}$ fraction of the optimal solution.

Example Assume for an even number k , there are $m = 3k/2$ advertisers and km items. Suppose these items are represented by an $m \times k$ matrix C . Each advertiser has a demand of k items. Advertiser i , $1 \leq i \leq k$, is only interested in items in row i , i.e. $\{c_{ij} | 1 \leq j \leq k\}$, and bids 1 for each of these items. Advertiser i , $k+1 \leq i \leq 3k/2$, bids $2 - \frac{k}{i+1-(k/2)}$

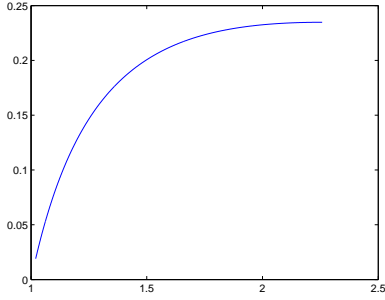


Figure 2: The bi-criteria approximation ratio of GREEDY for $1 < \alpha < \frac{\ln 2}{1 - \ln 2}$

for each item in row i . He also bids $2 - \frac{k}{i+1-(k/2)}$ for each item in column $i - k/2$.

By induction, one can show that at step j , for $1 \leq j \leq k/2$, GREEDY may allocate the top k items in column $k - j + 1$, to advertiser $3k/2 - j + 1$. Therefore, the total profit of GREEDY is:

$$k \sum_{i=k+1}^{3k/2} 2 - \frac{k}{i+1-(k/2)} = 2k \sum_{i=1}^{k/2} 1 - \frac{1}{1+(2i+1)/k}$$

We can make k arbitrary large. Hence, profit of GREEDY can be approximated by $k^2(1 - \ln 2)$, with arbitrary high accuracy:

$$\begin{aligned} k \sum_{i=k+1}^{3k/2} 2 - \frac{k}{i+1-(k/2)} &\approx k^2 \left(1 - \int_{x=0}^1 \frac{1}{1+x} dx\right) \\ &= k^2(1 - \ln 2) \end{aligned}$$

The optimal solution gives advertiser i , $1 \leq i \leq m$, all the items in row i . Hence, it gets a profit k^2 from the first k advertiser and a profit $k^2(1 - \ln 2)$ from the rest. Therefore, the approximation ratio of GREEDY is bounded by $\frac{1 - \ln 2}{2 - \ln 2}$.

5. AN ADAPTIVE GREEDY ALGORITHM

As suggested by Example 4.1, fixing the items allocated to each advertiser may hurt the performance of the algorithm. In this section, we give an adaptive greedy algorithm called ADAPTIVE GREEDY. Similar to the previous greedy algorithm, this algorithm iteratively selects new winners. However, in each iteration, it fixes only the *number* of items assigned to the new winner and not the precise *set* of items. This flexibility allows ADAPTIVE GREEDY to find the optimal solution in Example 4.1 and, as we will show, allows us to prove better approximation guarantees in general.

We use the following notation throughout this section. For a winning advertiser i , let c_i be the number of items allocated to i . Note as mentioned above that c_i remains fixed although the set of allocated items may change throughout the course of the algorithm. Let G_t be the set of advertisers chosen by the algorithm up to but not including time t . For $j \notin G_t$, let f_{jt} be the maximum number of items that we could allocate to advertiser j , subject to the constraint that each advertiser $i \in G_t$ gets exactly c_i items. More formally, f_{jt} is the solution of the following integer program.

$$\begin{aligned} &\text{maximize} && \sum_{k \in I_j} x_{jk} \\ &\text{subject to} && \sum_{k \in I_j} x_{jk} \leq d_j \\ &&& \forall i \in G_t : \sum_{k \in I_i} x_{ik} = c_i \\ &&& \forall k \in \mathcal{U} : \sum_{i \in \mathcal{A}} x_{ik} \leq 1 \\ &&& \forall i \in \mathcal{A}, \forall k \in \mathcal{U} : x_{ik} \in \{0, 1\} \end{aligned} \tag{9}$$

Also, let $r_{jt} = \frac{2f_{jt} - d_j}{f_{jt}} \cdot b_j$ be the maximum per-item profit that we could obtain from advertiser j , subject to the constraints. At step t , algorithm ADAPTIVE GREEDY chooses the advertiser with maximum r_{jt} , $j \notin G_t$. The algorithm is given in Figure 5.

To compute f_{jt} , one can solve the linear relaxation of the above program and round the solution to an integer solution (without decreasing the objective function). For a complete description of the algorithm, we refer the reader to [4].²

5.1 Performance Guarantee: Structural Approximation

As we observed in Section 3, the hardness result implies that we do not hope to analyze the adaptive greedy algorithm by the traditional multiplicative approximation framework. However, in order to distinguish the quality of different algorithms in practice, we can use other approximation measures such as the bi-criteria approximation.

In this section, we analyze the ADAPTIVE GREEDY algorithm using a newly defined framework for evaluating heuristics called *structural approximation* [10]. The main feature of this framework is that it is designed to evaluate the approximation ratio of an algorithm based on the *structure* of an optimal (or any) solution, and not only the value of an optimal solution. In this framework, given a solution with some value, we define the *recoverable value* of the solution based on its structure. The goal is to show that the value of our algorithm is at least as much as the recoverable value of *any* solution.³ We would like to design a recoverable value function that is very close to the real value. Note that in the traditional multiplicative approximation framework, the recoverable value is restricted to a multiplicative ratio of the value of the solution. By allowing more general functions for the recoverable value, the structural approximation framework is more flexible, and therefore can distinguish among different algorithms with the same worst-case multiplicative approximation guarantees. Additionally, this framework provides a way for comparing the quality of our solution to that of the optimal solution of the *same* problem (with the same parameters). This is in contrast to the bi-criteria approximation framework in which we compare the value of our solution to the value of a problem with different parameters. We refer the reader to [10] for a more detailed

²One can also solve the problem by modeling it as a flow problem with lower bound and upper bound on the capacities.

³Note that the recoverable value of different optimal solutions may differ. In fact, the solution with the optimum recoverable value may not be an optimal solution to the original problem at all. Our guarantees hold with respect to the largest recoverable value of any solution.

ADAPTIVE GREEDY Algorithm:

Initialize G , the set of chosen advertisers, to be the empty set.

Let t , the number of steps, be zero.

While there exists an unassigned advertiser:

1. $t \leftarrow t + 1$
 2. For each advertiser $j \notin G$:
 - (a) Let f_{jt} be the maximum number of elements that can be allocated to advertiser j , subject to the constraints that each advertiser $i \in G$ gets exactly c_i items.
 - (b) Let r_{jt} be the maximum profit-per-item that could be obtained by choosing advertiser j , subject to the constraints, i.e. $r_{jt} = (2 - d_j/f_{jt})b_j$.
 3. Choose an advertiser j with maximum r_{jt} .
 4. If r_{jt} is positive:
 - (a) Add advertiser j to G .
 - (b) Let $c_j = f_{jt}$.
- Otherwise terminate the algorithm.

Figure 3: ADAPTIVE GREEDY algorithm

discussion of the structural approximation framework.

We now analyze ADAPTIVE GREEDY algorithm using the structural approximation framework. Let \mathcal{W} be the set of advertisers that are allocated a non-empty set in the optimal solution, i.e., the winners. For each advertiser $i \in \mathcal{W}$, let S_i be the set of items allocated to i in an optimal solution. For $i \notin \mathcal{W}$, S_i is taken to be the empty set. Let $s_i = \frac{|S_i|}{d_i}$. Then the value of the optimal solution is: $\sum_{i \in \mathcal{W}} (2s_i - 1)d_i b_i$. We define

$$\sum_{i \in \mathcal{W}} (2s_i - 1 - \ln 2s_i)d_i b_i$$

as the recoverable value of this solution. Note that any optimal solution makes profit only from the advertisers with $s_i > \frac{1}{2}$. We show that the value of ADAPTIVE GREEDY is at least equal to this recoverable value. This gives a performance guarantee based on the structure of the optimal solution. In particular, the performance of the algorithm is better if s_i 's are closer to 1. This statement is formalized in the following theorem.

THEOREM 5.1. *Let S_i be the set of items assigned to advertiser i by an optimal solution. Also, let $s_i = \frac{|S_i|}{d_i}$. ADAPTIVE GREEDY algorithm obtains a value of at least:*

$$\sum_{i \in \mathcal{W}, s_i > \frac{1}{2}} (2s_i - 1 - \ln 2s_i)d_i b_i.$$

COROLLARY 5.1. *Consider a collection $\{(q_i, \alpha_i)\}$ corresponding to an optimal solution, where q_i is the fraction of the value of the solution comes from the advertisers for which an α_i fraction of the demand is satisfied. Then, the approximation ratio of ADAPTIVE GREEDY is at least:*

$$\frac{\sum_i q_i (2\alpha_i - 1 - \ln 2\alpha_i)}{\sum_i q_i (2\alpha_i - 1)}$$

In particular, when in the optimal solution, at least an s fraction of the demand of all winning advertisers is satisfied, ADAPTIVE GREEDY obtains at least a fraction $(2s - 1 - \ln 2s)$

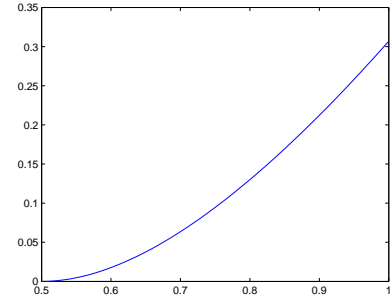


Figure 4: The approximation ratio of ADAPTIVE GREEDY when at least a s fraction of the demand of all winner advertisers is satisfied in the optimum solution.

of the value of the optimal solution. Figure 4 depicts this lower bound on the value for $\frac{1}{2} \leq s \leq 1$.

PROOF. To prove Theorem 5.1, without loss of generality, we assume the algorithm chooses advertisers $1, 2, \dots, T$ in order. We inductively construct a sequence of assignments of items to advertisers which we call canonical assignments. Let \mathcal{C}_0 be the empty set. The t -th element of the sequence of assignments, denoted by \mathcal{C}_t , is an assignment of items to advertisers $1, \dots, t$ which satisfies the following properties.

- Each advertiser i , $1 \leq i \leq t$, is assigned exactly c_i items.
- All items that are assigned in \mathcal{C}_{t-1} are also assigned in \mathcal{C}_t , i.e. are assigned to advertisers $1, \dots, t$.
- For each advertiser i , $1 \leq i \leq t$, all the items in S_i (namely, the items allocated to i by the optimal solution, if $i \in \mathcal{W}$) are assigned. (This property trivially holds for $i \notin \mathcal{W}$.)

In the following lemma, we show that such a sequence of assignments exists.

LEMMA 5.1. *There exists a sequence $\{C_1, C_2, \dots, C_t\}$ of canonical assignments which satisfies the properties above.*

PROOF. We prove the lemma by induction on t . For $t = 1$, the only nontrivial property is the third one, if $1 \in \mathcal{W}$. In this case we observe that $c_1 \geq |S_1|$, and hence a canonical assignment exists. Therefore, the base of induction holds.

Now consider $t > 1$. Assume a sequence of canonical assignments C_1, \dots, C_{t-1} , and let C_t be an optimal solution for LP (9). We change C_t to a canonical assignment satisfying the above three properties. For $1 \leq i < t$, let B_i be the set of item that are assigned to advertiser i in C_{t-1} . Also, let B_i the set of items in S_i that are not assigned to any advertiser in C_{t-1} . Note that $|B_i| \leq c_t$, because if we assign each item in B_i , $1 \leq i \leq t$, to advertiser i , it gives us a feasible solution of LP (9).

We repeat the following procedure as long as there exists an item x such that $x \in B_i$, for an advertiser $1 \leq i \leq t$, and x is not assigned to any advertisers in C_t . Since advertiser i has received c_i items in C_t , and $|B_i| \leq c_i$, there exists an item $y \notin B_i$ that is assigned to i in C_t . We update C_t by assigning x to i and removing y from the set of items allocated to i . Note that each time we perform the procedure, the number of items that is allocated to each advertiser remains the same. However, the number of items that are allocated to the same advertisers in C_t and the assignment defined by B_i 's increases by 1. Since the number of items that are allocated to the same advertiser in C_t and the assignment defined by B_i 's is at most $\sum_{1 \leq i \leq t} c_i$, after some number of steps, we reach a point where no such update is possible. Therefore, the final assignment after these set of updates satisfies all the three properties. \square

Now we are ready to prove Theorem 5.1. Note that the value obtained from each assignment only depends on the number of items allocated to each advertiser. Therefore, without loss of generality, we might assume that ADAPTIVE GREEDY allocates to each advertiser the same set of items assigned to this advertiser in canonical assignment C_T .

For $i \in \mathcal{W}$, let k_{it} be the number of items of S_i that are not assigned in C_{t-1} . Since we can construct a feasible solution with value k_{it} for LP (9) using C_{t-1} , we observe that $f_{it} \geq k_{it}$. Therefore, the value-per-item that ADAPTIVE GREEDY would obtain at this point if it chooses advertiser i and assigns to it all not previously assigned items of S_i at least $(2 - \frac{d_i}{k_{it}})b_i$. Thus, the value ADAPTIVE GREEDY obtains by the end of the algorithm is at least:

$$\sum_{i=1}^m \sum_{j=k_{iT}}^{|S_i|} (2 - \frac{d_i}{j})b_i$$

For $i \in \mathcal{W}$, if advertiser i is chosen by ADAPTIVE GREEDY then $k_{iT} = 0$, and if advertiser i is not chosen by ADAPTIVE GREEDY then necessarily $k_{iT} \leq \frac{d_i}{2}$. Therefore:

$$\sum_{i=1}^m \sum_{j=k_{iT}}^{|S_i|} (2 - \frac{d_i}{j})b_i \geq \sum_{i=1}^m \sum_{j=\lfloor d_i/2 \rfloor + 1}^{|S_i|} (2 - \frac{d_i}{j})b_i$$

By the same calculation as in Theorem 4.1, equation (6),

the value obtained by ADAPTIVE GREEDY is at least:

$$\sum_{i \in \mathcal{W}, s_i > \frac{1}{2}} (2s_i - 1 - \ln 2s_i)d_i b_i$$

Note that while we stated all our results with respect to the recoverable value of an *optimal* solution, our proofs in fact follow when considering the recoverable value of *any* solution.

5.2 Comparison of GREEDY and ADAPTIVE GREEDY

We analyzed ADAPTIVE GREEDY algorithm using the structural approximation framework. We also analyzed the GREEDY algorithm using the bi-criteria approximation framework. In order to compare these two greedy algorithms, we now analyze the GREEDY algorithm in the structural approximation framework.

Let \mathcal{W} be the set of the winners in an optimum solution (or any feasible solution). Assume that this solution satisfies a s_i fraction of the demand of advertiser $i \in \mathcal{W}$. Note that $s_i \geq \frac{1}{2}$, $i \in \mathcal{W}$. By equation (7), in the proof of Theorem 4.1, for any p , $0 \leq p \leq 1$, GREEDY obtains a value of at least:

$$\sum_{i \in \mathcal{W}, s_i > \frac{1}{2}} \min\{p, (1-p)(2s_i - 1 - \ln 2s_i)\}d_i b_i \quad (10)$$

Given s_i 's, one can compute the value of p which maximizes the sum and gives a recoverable value for each solution. However, by Theorem 5.1, the value of ADAPTIVE GREEDY algorithm from the same instance is at least:

$$\sum_{i \in \mathcal{W}, s_i > \frac{1}{2}} (2s_i - 1 - \ln 2s_i)d_i b_i$$

This immediately shows that ADAPTIVE GREEDY gives us a better guarantee on the value of the solution.

When in an optimum solution, fraction s of the demand of all advertisers is satisfied, we can easily compare the guarantees on the value given in Theorems 4.1 and 5.1. In this case, the optimum value for p in (10) is equal to $\frac{2s-1-\ln 2s}{2s-\ln 2s}$, which leads to the same guarantee on the approximation ratio of GREEDY algorithm. The guarantee on the approximation ratio of ADAPTIVE GREEDY algorithm is $2s - 1 - \ln 2s$. Figure 5 depicts these guarantees on the approximation ratio for $\frac{1}{2} \leq s \leq 1$.

One way to see that ADAPTIVE GREEDY strictly outperforms GREEDY is to observe that ADAPTIVE GREEDY algorithm obtains the optimum value of the instance described in Example 4.1. As a result, the ratio between the values of the solutions obtained by these two algorithms is $2 - \ln 2$.

6. CONCLUSION

In this paper we showed that having supply guarantees makes the problem of maximizing the value of selling banner advertisements inapproximable. In the lieu of this hardness result, we proposed two greedy heuristics for the banner advertising problem and analyzed them rigorously in the bi-criteria and structural approximation frameworks. Given the simple and flexible nature of these algorithms, we expect that these algorithms can help in automating the negotiation process for banner advertising.

In order to make our algorithms even more applicable to practical settings, we would like to consider the online

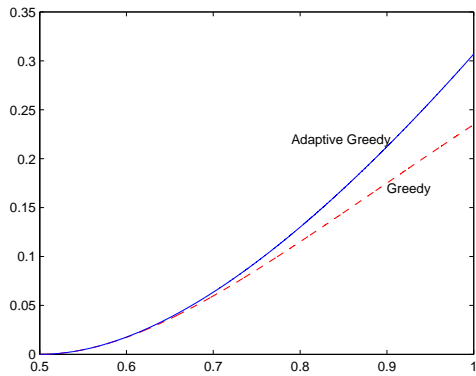


Figure 5: Comparison of the lower bounds on the competitive ratio of the two greedy algorithms when at least s fraction of the demand of all winners is satisfied by an optimal solution.

or stochastic settings, in which advertisers arrive over time and submit bids which must be immediately accepted or rejected, without knowledge of exact future demand. Also along these lines, it would also be useful to relax the assumption that the supply of items is known a priori. In practice, hosts have good estimates of the number of page views. However, these estimates may have large variance, and we should ideally provide algorithms which are robust with respect to errors in the supply estimates. Finally, to completely automate the banner advertising process and replace it with a repeated auction mechanism, it is necessary to design a pricing mechanism to accompany our allocation mechanism and analyze the resulting incentives.

7. REFERENCES

- [1] G. Aggarwal and J. D. Hartline. Knapsack Auctions. *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.
- [2] G. Ausiello, A. D’Atri, and M. Protasi. Structure Preserving Reductions Among Convex Optimization Problems. *Journal of Computer and System Sciences*, 21, 136-153, 1980.
- [3] S. Arora, S. Rao, and U. Vazirani. Expander Flows, Geometric Embeddings, and Graph Partitionings. *Proceedings of 36th Annual Symposium on the Theory of Computing (STOC)*, 222-231, 2004.
- [4] D. Bertsimas and J. Tsitsiklis. Introduction to Linear Optimization. *IIE Transactions*, 1998.
- [5] Y. Bartal, M. Charikar, and D. Raz. Approximating Min-Sum k -Clustering in Metric Spaces. *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, 2001.
- [6] Niv Buchbinder, Kamal Jain, and Seffi Naor. Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue. *Proceedings of the 15th Annual European Symposium on Algorithms (ESA)*, 2007.
- [7] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, Cambridge, MA, 2005.
- [8] S. Dobzinski, M. Schapira, and N. Nisan Truthful Randomized Mechanisms for Combinatorial Auctions. *Proceedings of the 38rd Annual ACM Symposium on Theory of Computing (STOC)*, 2006.
- [9] U. Feige. On Maximizing Welfare When Utility Functions Are Subadditive. *Proceedings of the 38rd Annual ACM Symposium on Theory of Computing (STOC)*, 2006.
- [10] U. Feige, N. Immorlica, V. Mirrokni, and H. Nazerzadeh. Structural Approximations - a framework for analyzing and designing heuristics. *manuscript*.
- [11] U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *Journal of ACM*, 45(4):634-652, 1998
- [12] U. Feige, V. Mirrokni, and J. Vondrak. Maximizing Non-Monotone Submodular Functions. *Proceedings of 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2007.
- [13] L. Fleischer, M. Goemans, V. Mirrokni, and M. Sviridenko. Tight Approximation Algorithms for Maximum General Assignment Problems. *Symposium of Discrete Algorithms (SODA)*, 2006.
- [14] U. Feige, L. Lovász and P. Tetali, Approximating Min Sum Set Cover, *Algorithmica* 40(4): 219-234, 2004.
- [15] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. WH Freeman & Co New York, 1979.
- [16] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial Auctions with Decreasing Marginal Utilities. *Proceedings 3rd ACM Conference on Electronic Commerce (EC)*, 2001.
- [17] D. J. Lehmann, L. O’Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *J. ACM*, 49(5), 577-602, 2002.
- [18] F. T. Leighton and S. Rao. An Approximate Max-Flow Min-Cut Theorem For Uniform Multicommodity Flow Problems with Applications to Approximation Algorithms. *Proceedings of 29th Annual Symposium on Foundations of Computer Science*, pages 422-431, 1988.
- [19] M. Mahdian, H. Nazerzadeh, and A. Saberi Allocating Online Advertisement Space with Unreliable Estimates *Proceedings 8th ACM Conference on Electronic Commerce (EC)*, 2007.
- [20] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. AdWords and Generalized Online Matching. *Journal of the ACM*, Volume 54 , Issue 5, 2007.
- [21] Pricewaterhouse Coopers LLP. IAB Internet Advertising Revenue Report. http://www.iab.net/resources/adrevenue/pdf/IAB.PwC_2006_Final.pdf
- [22] T. Roughgarden and É. Tardos. How Bad is Selfish Routing?. *Proceedings of IEEE Symposium on Foundations of Computer Science*, 93-102, 2000.
- [23] T. Sandholm. Algorithm for Optimal Winner Determination in Combinatorial Auctions. *Artificial Intelligence*, 135:1-54, 2002.
- [24] A. Schrijver. *Combinatorial Optimization*. Springer, 2003.
- [25] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2001.

APPENDIX

A. PROOF OF THEOREM 3.2

PROOF. We prove that claim holds even if the advertisers have large demands. We define a special case of selling banner advertisements. Consider m sets A_1, \dots, A_m , each of size k ; let $A = \cup_{i=1}^m A_i$ and $n = |A|$. Also consider a set Q of size q which is disjoint from A . Also, assume $k = O(q)$ and $kq = o(\sqrt{n})$.

Problem $\mathcal{B}(k, q, n)$ is defined with m advertisers. Let $c = 2(1 - \sqrt{\frac{q}{n} + \varepsilon})$. Let the bid of advertiser i , (I_i, d_i, b_i) , be equal to $(Q \cup A_i, ck, 1)$. We show that it is NP-Hard to approximate $\mathcal{B}(k, q, n)$ within a ratio of $2\sqrt{\frac{q}{n} + \varepsilon}$.

Consider the k -uniform regular set cover instances defined in the proposition 3.1, for ε and $s_0 = 3$. We say an instance is of type YES if all elements can be covered by t disjoint sets. An instance is of type NO if every $s \leq s_0 t$ sets cover at most a fraction of $1 - (1 - \frac{1}{t})^s + \varepsilon$ of the elements. We show if an algorithm can approximate problem $\mathcal{B}(k, q, n)$ within a ratio of $2\sqrt{\frac{q}{n} + \varepsilon}$, then this algorithm can distinguish between YES and NO instances.

For the YES instances, there are $t = \frac{n}{k}$ disjoint set which cover all the n elements. Hence, the value in this case is at least $t(k - (kc - k)) = n(2 - c)$.

For a NO instance, any s sets, $0 \leq s \leq s_0 t$, cover at most $q + (1 - (1 - \frac{1}{t})^s + \varepsilon)n$ elements. Therefore, the value of any s sets is at most $2(q + (1 - (1 - \frac{1}{t})^s + \varepsilon)n) - csk$. Note that t can be made arbitrary large ([14]); hence, we can approximate $(1 - \frac{1}{t})^s$ with $e^{-\frac{s}{t}}$ with arbitrary accuracy. Thus, the value of any s sets is at most:

$$2(q + (1 - e^{-\frac{s}{t}} + \varepsilon)n) - csk$$

We find the optimum value for s by the first order conditions:

$$2\frac{1}{t}e^{-\frac{s}{t}}n - ck = 0 \Rightarrow s = -t \log \frac{c}{2} \leq s_0 t \quad (11)$$

Plugging $s = -t \log \frac{c}{2}$ in (11), the maximum value is at most $2(q + (1 - \frac{c}{2} + \varepsilon)n) + cn \log \frac{c}{2}$. Note that the value for any choice of $m \geq s_0 t$ sets is negative, because the total demand is more than double of the number of elements. Therefore, the ratio between the value from a NO instance and a Yes instance is at most:

$$\frac{2(q + (1 - \frac{c}{2} + \varepsilon)n) + cn \log \frac{c}{2}}{n(2 - c)} = \frac{\frac{q}{n} + \varepsilon + 1 - \frac{c}{2} + \frac{c}{2} \log \frac{c}{2}}{1 - \frac{c}{2}}$$

$$\begin{aligned} &< \frac{\frac{q}{n} + \varepsilon + 1 - \frac{c}{2} + \frac{c}{2}(\frac{c}{2} - 1)}{1 - \frac{c}{2}} \\ &= 2\sqrt{\frac{q}{n} + \varepsilon} \end{aligned}$$

Recall that $1 - \frac{c}{2} = \sqrt{\frac{q}{n} + \varepsilon}$, which completes the proof. \square

B. FINDING A LOWER BOUND ON γ

Here, we compute a lower bound on the value of γ defined in (8). Taking the derivative of $\frac{2s_i - 1 - \ln 2s_i}{(\alpha + 1)s_i - \alpha}$ with respect to s_i :

$$\begin{aligned} &\frac{(2 - 1/s_i)((\alpha + 1)s_i - \alpha) - (2s_i - 1 - \ln 2s_i)(\alpha + 1)}{((\alpha + 1)s_i - \alpha)^2} \\ &= \frac{\alpha/s_i - 2\alpha + (\alpha + 1) \ln 2s_i}{((\alpha + 1)s_i - \alpha)^2} \quad (12) \end{aligned}$$

Define $f_\alpha(x) = \alpha/x - 2\alpha + (\alpha + 1) \ln 2x$, $x \in (0, 1]$. The first and the second derivative of f_α with respect to x are:

$$\begin{aligned} \frac{d f_\alpha}{d x} &= -\frac{\alpha}{x^2} + \frac{\alpha + 1}{x} \\ \frac{d^2 f_\alpha}{d^2 x} &= \frac{2\alpha}{x^3} - \frac{\alpha + 1}{x^2} \end{aligned}$$

Hence, for $\alpha > 1$ and $x \in (0, 1]$, function f_α is strictly convex and takes its minimum at $x = \frac{\alpha}{\alpha + 1}$. Therefore, f_α is increasing in $[\frac{\alpha}{\alpha + 1}, 1]$.

If $f_\alpha(1) = \ln 2 - (1 - \ln 2)\alpha \leq 0$, then f_α is negative in $[\frac{\alpha}{\alpha + 1}, 1]$. Hence, (12) is decreasing in $[\frac{\alpha}{\alpha + 1}, 1]$, and take its minimum at 1. For $\alpha \geq \frac{\ln 2}{1 - \ln 2}$, $f_\alpha(1) \leq 0$. Plugging in (8), $\gamma = \frac{2 - 1 - \ln 2}{(\alpha + 1)s_i - \alpha} = 1 - \ln 2$. Therefore, for this range of α , the approximation ratio is $\frac{1 - \ln 2}{2 - \ln 2}$.

For $1 < \alpha < \frac{\ln 2}{1 - \ln 2}$, as we prove later, $f_\alpha(\frac{\alpha}{\alpha + 1})$ is negative. Because $f_\alpha(1)$ is positive, by the intermediate value theorem, there exists a point x^* in $[\frac{\alpha}{\alpha + 1}, 1]$ for which f_α is zero. This point is the root of equation:

$$f_\alpha(x) = \alpha/x - 2\alpha + (\alpha + 1) \ln 2x = 0.$$

Because f_α is increasing in $[\frac{\alpha}{\alpha + 1}, 1]$, by convexity, x^* gives the minimum value of (12). Plugging this value for γ into $\frac{\gamma}{1 + \gamma}$, we get (5) as a lower bound on the approximation ratio of $1 < \alpha < \frac{\ln 2}{1 - \ln 2}$.

Now we only need to show that for $\alpha > 1$, $f_\alpha(\frac{\alpha}{\alpha + 1})$ is negative. For $x = \frac{1}{2}$, $f_\alpha(\frac{1}{2}) = 2\alpha - 2\alpha + (\alpha + 1) \ln 1 = 0$. Also, f_α is strictly convex and takes its minimum at $\frac{\alpha}{\alpha + 1} > \frac{1}{2}$. Therefore, $f_\alpha(\frac{\alpha}{\alpha + 1}) < 0$.