On the Power of Combinatorial Bidding in Web Display Ads

Khaled Elbassioni[1] and Mukesh Jha[2]

Masdar Institute
Abu Dhabi, UAE
Email: kelbassioni@masdar.ac.ae[1],
Email: mjha@masdar.ac.ae[2]

***Abstract*—Web display advertisement is occupying a major part of the marketing industry. With the vastly increasing number of competing advertisers, allocating web advertising space becomes a challenge, due to the need to run a large-scale auction in order to determine the winners and payments. As a result, some of the most desired properties, such as *truthfulness* (a.k.a. *strategy proofness*), and *social welfare maximization* are typically sacrificed by the auction mechanisms used in practice, in exchange of computational efficiency. Furthermore, those mechanisms typically assume a *fixed partition* of the advertising space, and use a simple mechanism, such as *Generalized Second Price (GSP)* auction to avoid the combinatorial explosion of the size of the problem when users are allowed to bid on *arbitrary regions of the space*. In this paper, we go beyond this non-combinatorial approach, and investigate the implementation of strategy–proof mechanisms which are truthful–in–expectation and approximately socially efficient, with an attempt to understand their computational efficiency, social welfare and revenue, when applied to Web display Ads. Those mechanisms were proposed recently as a theoretical engineering of the computationally less efficient mechanisms of Lavi and Swamy. Our experimental results show that allowing combinatorial bidding can offer substantial improvements in both social welfare and revenue as compared to slot-based methods, such as the GSP mechanism.**

***Keywords*—Combinatorial auctions; algorithmic mechanism design; Generalized Second Price; truthfulness.**

I. Introduction

Internet advertising is one of the most important marketing tools due to its growing number of audience. Internet advertising revenues in the United States totaled $42.8 billion for 2013, with an increase by 17% over the previous year [1]. One can basically distinguish two types of online advertisement: (I) Search advertising, which typically appears on search web pages, usually based on keywords searched by user, e.g., Google, Yahoo!; (II) Display advertising, which typically appears on non-search web pages, usually based on content and type of web-page, e.g., news sites, airlines sites, social networks sites, etc. [2].

In both types of advertisement, auctions are typically used to determine which advertisement will be displayed. In contrast to Search advertising, display advertising does not need a real-time auction mechanism and can thus be done offline, allowing for a substantially larger processing time. This paper will be concerned only with the second type.

Algorithmic mechanism design (AMD) studies optimization problems in which part of the input is not directly available to the algorithm; instead, this data is collected from self-interested agents. who can manipulate the algorithm by misreporting their parts of the input, if that would improve their own objective functions. It is therefore desirable to design a protocol or a *mechanism* which interacts with the agents so that their selfish behaviour yields a globally desirable outcome. Adding to this the requirement of *computational efficiency*, AMD quests for *efficient* algorithms that *(approximately) optimize* a global objective function (usually called *social welfare*), subject to the *strategic requirement* that the best strategy of the agents is to *truthfully* report their part of the input. Such algorithms are called *incentive compatible* or *truthful mechanisms*.

If the underlying optimization problem can be efficiently solved to optimality, the celebrated *VCG mechanism* [3] achieves truthfulness, social-welfare optimization, and polynomial running time. In general, and more specifically in the display Ad auctions with a relatively large number of advertisers, the underlying optimization problem can only be solved approximately. Lavi et al. [4][5] showed that certain linear programming based approximation algorithms for the social welfare problem can be turned into randomized mechanisms that are truthful-in-expectation, i.e., reporting the truth maximizes the expected utility of an agent. The Lavy-Swamy (LS)-reduction is powerful [4]–[7], but unlikely to be efficient in practice because of its use of the *Ellipsoid method* for linear programming. In fact, we are not aware of any attempt to apply the LS-approach in practice or at least to perform a systematic study of its applicability and effectiveness, compared to the mechanisms which are currently being used.

Presently Google and Yahoo! are using Generalized-Second-Price (GSP) auction mechanism to auction off slots. GSP looks somewhat similar to VCG but its properties and equilibrium behavior are quite different. Unlike the VCG mechanism, GSP is *not* truthful [8], but is by far computationally more efficient. On the other hand, a major drawback of GSP when applied to Display Ad auctions is that it is inherently *slot-based*, that is, the advertising space has to be apriori partitioned into fixed slots, which are auctioned off in a way similar to keyword auctions. This has the obvious disadvantage of limiting the bidding power of the agents, which could be otherwise exploited to increase the total social welfare of the agents and/or the revenue due to the auctioneer. Allowing combinatorial bidding, where agents can bid on different regions (bundles) of the advertisement space can offer substantial improvements in both social welfare and revenue, provided that there are computationally efficient mechanisms that can deal with this kind of bidding.

Very recently, Elbassioni et al. [9] gave an efficient implementation of the LS-approach based on the simpler *multiplicative weights update (MWU) methods*. The simplification comes at the cost of slightly weaker approximation and truthfulness guarantees. In this paper, we investigate the effectivity of their method when applied to display advertisement auctions. We carry out extensive experiments to compare these methods with GSP in terms of truthfulness, social welfare and revenue. Our experiments show that the proposed implementation can handle auctions involving hundreds of bidders and thousands of bundles, and offer substantial improvements in both social welfare and revenue as compared to slot-based methods, such as the GSP mechanism.

In Section II, we discuss about Internet advertisement, auction, drawbacks of non-truthful auction mechanisms and its impact on online advertisement. In Section III, we explain the outline of the problem and algorithmic solutions. We also define relevant terminology and definitions. In IV, we elaborate our experimental set-up. We discuss our evaluation process in Section V. We present our results and analysis for social-welfare, revenue and running time in Section VI. We compare the VCG-based mechanism with GSP mechanism in Section VII. Finally we conclude our work in Section VIII.

II. Related Work

Online advertisement dates back to at least 1994 when HotWire started to sell banner ads to advertisers [10]. After a small hiatus during the period of dot-com crash, online advertisement along with online auctions became one of the major form of exchange of items and services over the Internet. GoTo.com introduced the use of auction mechanisms for allocating advertising space on web pages showing results of the query to generate revenue [10]. Presently, most of online advertisement uses some form of auction based on keywords, combination of keywords, or space [8][10][11].

Overture (now a part of Yahoo!) operated a first-price auction in early 2000s for search advertisement. Since this form of auction is not truthful, bidders used to bid strategically to increase their utility [12]. Furthermore, there was a loss in revenue and the market was unstable due to frequently lying bidders. Hence, VCG-based mechanisms were suggested to stabilize auction outcomes. Since, these mechanisms require solving an NP-hard winner determination problem, a variant of second-price auction came in practice, the so-called Generalized Second Price (GSP) auction. Yahoo! and Google AdWords use the GSP mechanism. This mechanism is non-truthful for multiple items, but it has envy-free equilibrium [8]. Although it is better than the first-price auction and widely used, various researches showed that the GSP mechanism has its own flaws. In particular, the bidders maybe forced to undertake complicated tasks of choosing a bidding strategy to increase their utility. Asdemir [13] and Edelman et al. [12] showed that the GSP mechanism might result in bidding wars cycles and static bid patterns are frequently observed. The strategy of bidders to outbid each other until one of them drops their bid and the other one follows by dropping its bid to just by very small ϵ above the competitor's bid is known as Bidding war cycle. Static bid patterns is defined as the bidder's fixed pattern of bidding based on their expectation to win. This might result in unstable markets. Matthew et al. [14] showed that for some strategic bidding the GSP mechanism does not converge for 3 or more slots/items. These results show that GSP is not strategy-proof and there is a requirement for strategy proof mechanisms for a stable market.

The term AMD was first coined by Nisan et al. [15] in 1999. AMD combines the idea of utility maximization of independent selfish agents and mechanism design from economics, Nash equilibrium and individual rationality from game theory, with the analytic tools of Theoretical Computer Science, such as computational constraints, worst-case analysis and approximation ratios which are not addressed in classical economic mechanism design. From practical implementation point of view, the most important aspect of AMD is the analysis of computational efficiency. If a mechanism cannot be implemented to scale well with respect to number of items and bidders, it cannot be considered as a viable solution. This rules out many classic economic mechanisms which satisfy the mechanism designer's requirements but are computationally inefficient in general to implement. The celebrated classical mechanisms like, Vickrey-Clarke-Groves auction (VCG) and Generalized Vickrey Auction, involve the solution of NP-hard winner determination problems and, in spite of their rich game theoretic properties, are impractical to implement.

Combinatorial auctions are market mechanisms in which bidders can bid on bundles of multiple items, i.e., combination of items. The common example of combinatorial auctions are Federal Commission for Communications (FCC) auctions of spectrum licenses, course registration, airport take-off and landing time slots, job shop scheduling and electricity markets etc [16]. There are various mechanisms proposed for combinatorial auctions, such as AUSM auction, RAD mechanism, PAUSE mechanism and iBundle mechanism etc [16].

Sandholm [17] presented an approximate search algorithm for solving combinatorial auctions. He showed that dynamic programming and exhaustive enumeration methods are too complex to scale for large number of items (20-30 items). Restricting the combinations might result in polynomial time winner determination problem for combinatorial auctions but it has economic inefficiency, since imposing restrictions on certain combinations of items bars bidders from bidding on the combination they might prefer. Furthermore, the bids in [17] were generated by randomly taking values from either $[0; 1]$ or from $[0; g]$, where g is the number of items in the bid. This method of bid generation does not consider various economic aspects, such as highly valued/preferred combinations, sub-additive bids, super-additive bids, etc. Hence, [18] claimed that this type of bid generation results in computationally trivial winner determination problem.

In [19], an optimal auction mechanism for multi-unit combinatorial auctions for single-minded bidders was presented. If a bidder is only interested in a single specified bundle (or any of its supersets), it is known as single-minded bidder; a single-minded bidder values at zero any other (non-superset) bundle. In real life, a bidder might be interested in buying multiple bundles with varying preferences. Thus, this setting is not practically applicable in real life. Furthermore, [19] does not consider *sub-additive* and *super-additive* scenarios for combinatorial auctions bids. Archer et al. [20] gave an approximate truthful *randomized* mechanism for combinatorial auctions with single parameter agents. In a single parameter setting, bidders provide one single input to specify their preference, e.g., single-minded bidder. They provided a general technique to modify linear programming-based randomized rounding algorithms to make them monotone with high probability, giving an efficient truthful-in-expectation (TIE) mechanism that approximately maximizes social-welfare. However, single parameter mechanisms are rarely used in practice [21], as the present development in online transactions, Internet markets, Internet advertising and online auctions mandate multi-parameter setting mechanisms. Lavi et al. [4][5] extended the results in [20] to non-single parameter settings and showed that certain linear programming based approximation algorithms for the social welfare problem can be turned into randomized mechanisms that are truthful-in-expectation with the same approximation ratio. Dughmi et al. [22] suggested the first black-box reduction from arbitrary approximation algorithms to truthful approximation mechanisms problems for a non-trivial class of multi-parameter problems. In particular, they proved that every social-welfare-maximization problem that admits an FPTAS and can be encoded as a packing problem, also admits a truthful-in-expectation randomized mechanism which also an FPTAS.

In display Ad auctions, the social welfare maximization problem amounts to finding a maximum-weight independent set (that, is a pairwise-disjoint collection) of squares (or more generally ”fat” rectangles), among a given set of weighted squares. Auctions of this type have been considered in [23][6], but only theoretical results were provided without considering the efficient implementation of the proposed mechanisms.

In this paper, we will focus on the efficient implementation of Lavi-Swamy mechanism, as proposed in [9], and its application to display Ad auctions as suggested in [6]. To the best of our knowledge, this is the first attempt to implement a VCG-based mechanism, which scales with the number of bidders and items.

III. The Setting

A. Items and Bundles

The Advertising space is divided into small units of unitary squares which we refer to as *items*. A combination of items is called a *bundle*. In our case, bundles are restricted to be also squares; see Figure 1 where bundles are represented by dotted squares.

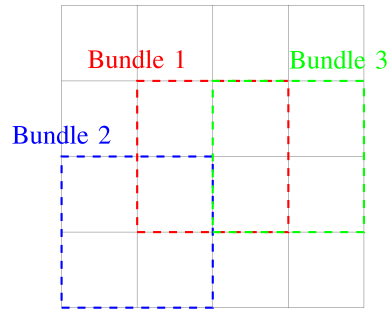

Figure 1. Items and bundles

B. Bidding Mechanism

This refers to the manner in which the value of bid for items and bundles are offered or quoted.

We distinguish the following types of valuations (bids) v_i, for bidder i:

1) Sub-additive setting: For a bundle S consisting of unitary squares $S_1, \ldots, S_k$, $v_i(S) \leq \sum_{j=1}^{k} v_i(S_j)$. This takes into account the bundles/items that are substitutabilities.
2) Super-additive setting: For a bundle S consisting of unitary squares $S_1, \ldots, S_k$, $v_i(S) \geq \sum_{j=1}^{k} v_i(S_j)$. This takes into account of bundles/items that are complementarities.
3) Arbitrary-setting: In arbitrary case, we provide bidders the capability to choose between sub-additive and super-additive valuations.

We call a bidder k-*minded* if (s)he bids a positive value on at most k bundles.

C. The Social Welfare Maximization Problem

Given the vector of bids $v = (v_1, \ldots, v_n)$, the social welfare maximization (SWM) problem is to find the optimum *integer* solution of the following linear program:

$$z^*(v) = \max \sum_{i,S} v_i(S) x_{i,S} \tag{1}$$

$$\text{s.t.} \sum_{i,S:\ j \in S} x^*_{i,S} \leq 1 \qquad \forall \text{ items } j \tag{2}$$

$$\sum_{S} x^*_{i,S} \leq 1 \qquad \forall \text{ bidders } i \tag{3}$$

$$x_{i,S} \geq 0 \qquad \text{for all } i, S.$$

Informally, we want to find an allocation that maximizes the total valuations (i.e., the social welfare) while making sure that (2) each item is only assigned to one bidder and (3) each bidder is only assigned at most one bundle. In our implementation, we used CPLEX [24], to solve the relaxation LP.

For $\alpha \in (0, 1]$, we say that an algorithm $\mathcal{A}$ is an α-*integrality-gap-verifier* for the LP (4) if for any vector of bids v and any (fractional) feasible solution x^* of the LP, $\mathcal{A}$ returns an *integral* solution x^I of social welfare $\sum_{i,S} v_i(S)x^I_{i,S} \geq \alpha \cdot \sum_{i,S} v_i(S)x^*_{i,S}$. A randomized $\frac{1}{16}$-integrality gap verifier for (4) was given in [6] and works as follows:

Algorithm 1 Integrality-Gap-Verifier (x^*)

Require: A fractional allocation x^*
Ensure: An integral solution x^I s.t. $\sum_{i,S} v_i(S)x^I_{i,S} \geq \frac{1}{16}\sum_{i,S} v_i(S)x^*_{i,S}$

1: **for** each bidder i **do**
2: choose a bundle S_i as follows

$$S_i = \begin{cases} S & \text{with prob. } \frac{1}{8}x^*_{i,S} \\ \emptyset & \text{with prob. } 1 - \frac{1}{8}\sum_S x^*_{i,S} \end{cases}$$

3: **end for**
4: Let $W = \emptyset$; $x^I = \mathbf{0}$
5: Let $S_1, S_2, \ldots, S_\ell$ be the bundles selected in Step 2 in *non-increasing* order of size
6: **for** $i = 1, \ldots, \ell$ **do**
7: **if** S_i does not intersect any range S_j with $j \in W$ **then**
8: add i to W; $x^I_{i,S_i} = 1$
9: **end if**
10: **end for**
11: **return** x^I

In our implementation, we run Algorithm 1 a constant number of times and take the solution with largest social welfare, to ensure with high probability that it returns a solution with social welfare at least $\frac{1}{16}$ of the optimal.

D. Randomized Truthful-in-expectation Mechanisms

A mechanism consists of an allocation rule and a payment scheme. Given the vector of bids v, the allocation rule determines a feasible allocation, that is, a feasible integral solution to the LP (4), while the payment scheme determines the payment p_i to be charged to bidder i. The utility of a bidder is defined as her valuation over her allocated bundle minus her payment: $U_i(v) := v_i(S) - p_i$. In a randomized mechanism, all these (allocation, payments, and utility) are random variables. Such a mechanism is said to be truthful-in-expectation (TIE) if for all i and all $v_i, \bar{v}_i, v_{-i}$, it guarantees $\mathbb{E}[U_i(\bar{v}_i, v_{-i})] \geq \mathbb{E}[U_i(v_i, v_{-i})]$, where the expectation is taken over the random choices made by the algorithm. Here, we denote by $\bar{v}_i$ the *true* valuation of bidder i, and use the standard terminology $v_{-i} := (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$. In other words, the expected utility is maximized when the bidder reports her valuation truthfully.

E. The Lavi-Swamy Mechanism

We next review the LS-reduction. It consists of three steps:

1) Find an optimal solution x^* for the LP-relaxation (4), and determine the VCG prices p_1 to p_n. The price for the i^{th} agent is $p_i = \sum_{j \neq i}\sum_S v_j(S)(\hat{x}_{j,S} - x^*_{j,S})$, where $\hat{x}$ is an optimal solution for LP (4) with input $(0, v_{-i})$, that is, when bidder i is removed from the auction. The allocation x^* and the VCG-prices are a truthful mechanism for the fractional problem.
2) Write $\alpha \cdot x^*$ as a *convex combination of integral solutions*, i.e., $\alpha \cdot x^* = \sum_I \lambda_I x^I$, $\lambda_I \geq 0$, $\sum_{I \in \mathcal{N}} \lambda_I = 1$, and x^I is an integral solution to (4). This step requires an *α-integrality-gap-verifier* for (4) for some $\alpha \in (0, 1]$.
3) Pick the integral solution x^I with probability λ_I, and charge the i-th agent the price[1] $p_i \cdot (\sum_S v_i(S)x^I_{i,S} / \sum_S v_i(S)x^*_{i,S})$.

F. The EMR Implementation

If one considers the implementation of the LS-mechanism in the display Ad setting, step 2 stands as the major bottleneck as it requires solving a linear program with an *exponential* number of variables. A direct solution of this would require the use of the Ellipsoid method for linear programming which is typically highly inefficient in practice. Elbassioni et al. [9] proposed a solution using the simpler multiplicative weights update methods, which were used for speeding-up convex optimization [25]–[31]. In particular, it was shown that a variation of the approach by Garg et al. [27] can be used to obtain a convex combination that dominates $\alpha \cdot x^*$. Then the packing property of the polytope can be used to covert this into an exact equality algorithm (see Algorithm 3 below). The details of this are given in the following sections.

1) Finding a Dominating Convex Decomposition: Such a decomposition is equivalent to finding an optimal solution of the following LP:

$$\min \sum_I \lambda_I \tag{4}$$

$$\text{s.t.} \sum_I \lambda_I x^I_{i,S} \geq \alpha \cdot x^*_{i,S} \quad \text{for all } (i,S) \in L \tag{5}$$

$$\sum_{x \in \mathcal{N}_I} \lambda_I = 1 \tag{6}$$

$$\lambda_I \geq 0 \quad \text{for all } I.$$

Here, $L = \{(i,S) : \; x^*_{i,S} > 0\}$.

By turning (C2) into an inequality $\sum_{x \in \mathcal{N}_I} \lambda_I \geq 1$, the authors in [9] reduced the problem to a *covering* linear program, which can be solved via the approach in [27][28], at the cost of losing a factor of $(1 + 4\varepsilon)$ in the approximation ratio. The procedure is given as Algorithm 2 and works by maintaining a set of weights $p_{i,S}$ that can be though of as a penalty for the violation in the constraint corresponding to the pair (i,S) in (C1). As long as a scaled version of (C1) is not satisfied for some (i,S), the algorithm uses the weights $p_{i,S}$ (in step 5) to construct a valuation vector v' that is fed to the α-integrality gap verifier $\mathcal{A}$ in step 6, which in turn returns an integral solution x^I. This solution and its multiplier λ_I (computed in steps 7-9) are added to the list of solutions $\mathcal{I}$. Then the set of "active" constraints L (which are not yet satisfied) is updated and a new iteration is started.

For $y \geq 0$, we define the function

$$h(y) = \begin{cases} y & \text{if } y < 1 \\ -\infty & \text{otherwise.} \end{cases}$$

[1] If $\sum_S v_i(S)x^*_{i,S} = 0$, the price is zero.

Algorithm 2 Dominating$(x^*, \mathcal{A})$

Require: A feasible fractional solution to (4), an α-integrality gap verifier $\mathcal{A}$ for (4), and an accuracy parameter $\varepsilon \in (0, 1/2]$
Ensure: A collection of integral feasible solutions $\{x^I\}_{I\in\mathcal{I}}$ to (4) and convex multipliers $\{\lambda_I\}_{I\in\mathcal{I}}$ s.t. $\sum_{I\in\mathcal{I}} \lambda_I x^I \geq \frac{\alpha}{1+4\varepsilon} x^*$

$\mathcal{I} := \emptyset$; $M = 0$
$L := |\{(i,S):\ x^*_{i,S} > 0\}|$; and $T := \frac{\ln |L|}{\varepsilon^2}$
while $M < T$ **do**
  $p_{i,S} := (1-\varepsilon)^{h(\sum_{I\in\mathcal{I}} \frac{\lambda_I x^I_{i,S}}{\alpha x^*_{i,S}})}$, for $(i,S) \in L$
  Set
$$v'_i(S) := \begin{cases} \frac{p_{i,S}}{\alpha x^*_{i,S} \cdot \sum_{i',S'} p_{i',S'}} & \text{for } (i,S) \in L \\ 0 & \text{otherwise.} \end{cases}$$
  Let $x^I := \mathcal{A}(x^*, v')$;
  $\lambda_I := \min_{(i,S)\in L:\ x^I_{i,S}=1} \alpha x^*_{i,S}$
  if $\sum_{I'\in\mathcal{I}} \lambda_{I'} < T$ **then**
    $\lambda_I := \min\{\lambda_I, 1\}$
  end if $\mathcal{I} := \mathcal{I} \cup \{I\}$
  $L := L \setminus \{(i,S):\ \sum_{I\in\mathcal{I}} \frac{\lambda_I x^I_{i,S}}{\alpha x^*_{i,S}} \geq T\}$
  $M := \min\left\{\min_{(i,S)\in L} \sum_{I\in\mathcal{I}} \frac{\lambda_I x^I_{i,S}}{\alpha x^*_{i,S}}, \sum_{I\in\mathcal{I}} \lambda_I\right\}$
end while
$\lambda_I := \lambda_I / \sum_{I'\in\mathcal{I}} \lambda_{I'}$, for $I \in \mathcal{I}$
return $(\{x^I\}_{I\in\mathcal{I}}, \{\lambda_I\}_{I\in\mathcal{I}})$

At the end of the procedure, the λ_I's are normalized to get a collection of integral feasible solutions $\{x^I\}_{I\in\mathcal{I}}$ to (4) and convex multipliers $\{\lambda_I\}_{I\in\mathcal{I}}$ s.t.

$$\sum_{I\in\mathcal{I}} \lambda_I x^I \geq \frac{\alpha}{1+4\varepsilon} x^*; \tag{7}$$

see [9] for details.

2) Getting an Exact Decomposition: Given the dominating convex decomposition (7), Algorithm 3 can be used to modify it into an exact decomposition. The idea is to use the *packing* property of the feasible set[2] to add more feasible solutions with smaller convex multipliers to offset the difference between the L.H.S and R.H.S. of (7).

We refer the interested reader to [9] for the details as well as the running time analysis of Algorithms 2 and 3, and only summarize the results here.

Theorem 1. *Consider a Display Ad auction on n k-minded bidders and m items. Then, for any $\varepsilon > 0$, there is a TIE which approximates the optimum social welfare within a factor of $\frac{1}{16}(1-4\varepsilon)$ and whose running time is $O(n \cdot T_{LP}(n(k+m), n+m, V_{\max}) + \frac{n^2 m(n+m)\log(n+m)}{\varepsilon^2})$, where $T_{LP}(\ell, r, V_{\max})$ is the time to solve an LP (4) with ℓ variables, r constraints and maximum objective function coefficient $V_{\max} := \max_{i,S} v_i(S)$.*

[2]that is, if x is feasible solution then any $\mathbf{0} \leq \mathbf{x'} \leq \mathbf{x}$ is also feasible.

Algorithm 3 Equality$(\widehat{x}, \{x^I\}_{I\in\mathcal{I}}, \{\lambda_I\}_{I\in\mathcal{I}})$

Require: A feasible solution $\widehat{x}$ to (4), a collection of integral feasible solutions $\{x^I\}_{I\in\mathcal{I}}$ to (4) and convex multipliers $\{\lambda_I\}_{I\in\mathcal{I}}$ s.t. $\sum_{I\in\mathcal{I}} \lambda_I x^I \geq \widehat{x}$
Ensure: A collection of integral feasible solutions $\{x^I\}_{I\in\mathcal{I}}$ to (4) and convex multipliers $\{\lambda_I\}_{I\in\mathcal{I}}$ s.t. $\sum_{I\in\mathcal{I}} \lambda_I x^I = \widehat{x}$

Create a new integral solution $x^{I_0} = y^0 := \mathbf{0}$
$\mathcal{I} := \mathcal{I} \cup \{I_0\}$; $\lambda_{I_0} := 0$
while $\exists I \in \mathcal{I}, (i,S)$ s.t. $x^I_{i,S} = 1$ and $\sum_{I\in\mathcal{I}} \lambda_I x^I_{i,S} - \lambda_I \geq \widehat{x}$ **do**
  $x^I_{i,S} := 0$
end while
while $\exists (i,S) : \Delta_{i,S} := \sum_{I\in\mathcal{I}} \lambda_I x^I_{i,S} - \widehat{x}_{i,S} > 0$ **do**
  Let I be s.t. $x^I_{i,S} = 1$
  $B := \{(i',S'):\ x^I_{i',S'} = 1 \text{ and } \Delta_{i',S'} > 0\}$; $b = |B|$
  Index the set of pairs $\{(i,S)\}_{i,S}$ s.t. $B = [1..b]$ and $\Delta_1 \leq \ldots \leq \Delta_b$
  For $\ell \in [0..b-1]$ define a vector y^ℓ by
$$y^\ell_j = \begin{cases} 1 & \text{for } j \leq \ell, \\ 0 & \text{for } j > \ell \end{cases}$$
  $\lambda_I := \lambda_I - \Delta_b$
  for $1 \leq \ell < b$ **do**
    Create a new integral solution $x^{I'} := y^\ell$
    $\mathcal{I} := \mathcal{I} \cup \{I'\}$; $\lambda_{I'} := \Delta_{\ell+1} - \Delta_\ell$
  end for
  $\lambda_{I_0} := \lambda_{I_0} + \Delta_1$
end while
return $(\{x^I\}_{I\in\mathcal{I}}, \{\lambda_I\}_{I\in\mathcal{I}})$

IV. Experimental Setup

A. Data Set

Since general combinatorial auctions have never been widely implemented, it is hard to find realistic data sets for them. In view of this, it is natural to rely on artificially generated data sets which can capture the economic and combinatorial issues and successfully represent the sort of scenarios one might expect to encounter [18][32] [33].

The first ever known attempt for generating test data sets for combinatorial auctions (CAs) was analyzed by Leyton-Brown et al. in [18]. They discussed guidelines for generation of CA's bid data and provided a method for bid generation which can successfully capture the economic and underlying structural factors of items and their combinations in CAs. Although they considered as many factors as possible, they overlooked a number of economic issues which should be considered for CA bid data generation [33].

For our experimental data, we considered various aspects as suggested in the literature for generating the data. The aim of generated data was to capture real-life competitive bidding along with ideas of economic substitutability and complementarity of items. The items, bundles and bidders were randomly generated. The bidding value on each item/bundle was made competitive by enforcing a range on bid values of each item such that bidders can randomly bid within that range. Furthermore, the users were allowed to bid on bundles of items, and the bid values of the bundles could be sub-additive, super-additive or arbitrary.

The following aspects were considered in data generation to capture economic parameters of realistic CAs:

1) *Number of bidders* n: The number of bidders is a good measure of the size of the problem. It essentially depends on whether the auction is between firms or individuals. For firms, a realistic number is substantially $n < 100$ [33]. In case of individuals, n can be up to 10,000 for a typical business-to-consumer auction [33].
   For our experimental setup, we considered $50 \leq n \leq 400$. Further, we also considered the number of bidders relative to number of items (m), reasonably being $0 < \frac{n}{m} \leq 2$.
2) *Number of items* m*, number of bundles* k: For m unique items, there are $O(m^2)$ possible bundles (each is a square region). This shows the input complexity of CAs. In realistic settings, not all combinations are preferred and the total number of bundles is $\ll O(m^2)$, as there are certain combinations which are preferred by most of the agents. In case of advertisement auctions, typically central regions of the screen, are more valued by bidders. We imposed this restriction in our experimental setting. In our experiments, the number of items/bundles ranges from a few dozens to a few hundreds. In particular, we considered k-minded bidders for $k \in [0, 200]$. In realistic settings, the total number of items $m > 100$ [33].
3) *Bidders valuations* v_i: In realistic settings, bidders go for competitive bidding, i.e., they have a speculation of the possible value of the item or combination of items which is close to its market value, and it is likely that they bid within a certain range of the market value [33]. To capture this idea we associated a range with each item, and bidders can randomly bid within that range for the item.
   If the set of items are structured, other considerations also come into picture, such as location. Our case of online auction also represents such a case. The assumption that all bundles of the same size are equally likely to be requested is clearly violated in real-world auctions [18]. Most of the time, advertisers would like to get slots or combination of slots at particular location or be indifferent to certain locations.
   The following are important parameters that affect the bidding mechanism:
   - Stretch Factor (S-factor): To make the bidding competitive, an S-factor is associated with each item/bundle such that bidders can randomly bid within the stretch factor of the item/bundle value. This captures the real-life setting where bidding values of bidders for a particular item remain close to the estimated or assumed market value.
   - Additive/Subtractive factor (E-factor): To regulate the bid price in case of bundles, we associated an E-factor to the bid value of bundles. The E-factor determines the factor of value that can be added (super-additive) or subtracted (sub-additive) from the total sum of bids of individual items in the bundle.

   If bidding values are not regulated or chosen carefully, then even a hard distribution can become computationally easy [18]. For example, if one particular bidder is randomly bidding very high as compared to others, it can make the optimization problem an easy matching and does not capture the idea of competitive bidding [18]. Boutilier et al. [34] considered the values of bids from the normal distribution with mean 16 and standard deviation 3 and Sandholm [17] generated bid values randomly from either $[0, 1]$ or from $[0 : g]$, where g is the number of items in the bid. Since the bid values are not related to number of items in a bundle, these methods are unreasonable (and computationally trivial) [18].
   In our experimental setting, we provided a stretch factor (S-factor) which ensured that the bid value for bigger bundle is within a certain range of the cumulative sum of bids of smaller bundles in the bigger bundle. We considered the super-additive and sub-additive cases along with the arbitrary case.

V. Evaluation Process

We evaluated our algorithm in terms of truthfulness, running time, social welfare and revenue generated.

A. Truthfulness

To test experimentally the truthfulness of the VCG-based or the GSP-mechanisms, we modified the bid of a particular agent to see if (s)he can increase its expected utility by lying. In particular, we conducted experiments such that one particular, randomly selected bidder is lying, i.e., changes her (his) bid values in small increments while all other bidders fix their bid values. We computed the maximum utility the lying bidder achieves by the mechanism, and computed the ratio to the actual utility that bidder would have got by truthfully bidding. We took the average ratio over thousands of experiments and used this as a measure of how truthful is the mechanism.

B. Running Time

In the display Ad setting, where hundreds of agents are typically participating simultaneity in the auction, it is mandatory for an auction designer to consider its running time. The practical application of any mechanism depends crucially on the fact that it should meet the time requirement.

We have considered the running time of our VCG-based mechanism with respect to increasing the number of items/bundles and the number of bidders.

C. Social Welfare

Theorem 1 states that our implemented mechanism ensures that the resulting allocation is almost $\frac{1}{16}$-socially efficient, i.e., the resulting social welfare is almost at least $1/16$ times the optimal social-welfare. While the GSP-mechanism does not ensure a similar guarantee, we experimentally tested the gap in social welfare between the two mechanisms. Since GSP does not allow combinatorial bidding, we expect the behavior to be dependent on the type of valuations used; for supper-additive (resp., sub-additve) valuations, the mechanism VCG-based mechanism would perform better (worse) than GSP with respect to social welfare.

D. Revenue

Revenue can be defined as the amount of value generated by the auction in the market which can be taken by seller. It is, in essence, the summation of all the bidders' payments made to the seller. One of the important aspects of any auction mechanism is also the revenue generated.

VI. Results And Analysis

In this Section, we analyse the truthfulness, running time, revenue, and social welfare of the VCG-based mechanism. We further present results comparing this mechanism to GSP. Our experimental results show that the VCG-based mechanism is superior in many aspects to the GSP mechanism. In this Section, we mainly consider the setting where $0.001 <$ S-factor < 0.1 and $1 <$ E-factor < 100 and grid size is 10×10, unless otherwise stated. The vertical bar shows the 95% confidence in the results obtained.

A. VCG-based Mechanism analysis

We conducted our experiments with CPLEX [24] and approximate packing algorithm. All the experiments in this Section were run independently for 100 times and their average was calculated for every-parameter.

1) Social-Welfare: We observed that the social welfare obtained increases with the number of bidders and then comes to a saturation value for super-additive and sub-additive case, as shown in Figure 2.

2) Revenue: The revenue generated by our mechanism showed its applicability in real life as shown in Figure 3. Although the VCG-mechanism does not theoretically provide any guarantee on revenue, our experiments show that it provided a good revenue for the market maker. Any commercial auction market cannot be subsided, so it must generate enough revenue for itself to sustain.

3) Running Time: As Theorem 1 claims, the running time for our mechanism is acceptable with respect to number of bidders, items, and bundles. An empirical verification of this can be seen in Figure 4. As the number of bidders increases, the running time also increases almost linearly. We observe that the running time mostly depends on the amount of fractional components in the fractional allocation; if this number is high, the time needed to decompose the fractional allocation into an integral one increases.

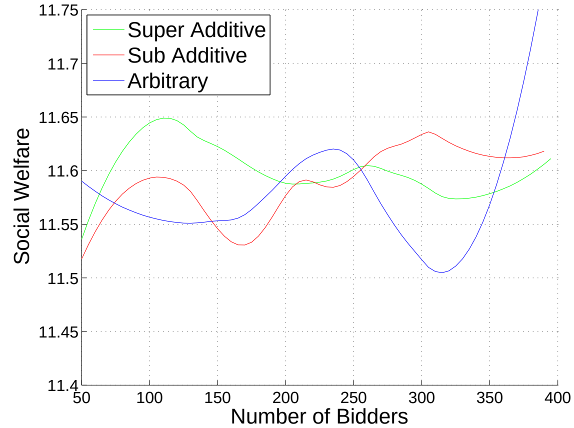

Figure 2. Log curve for Social Welfare of VCG-based mechanism

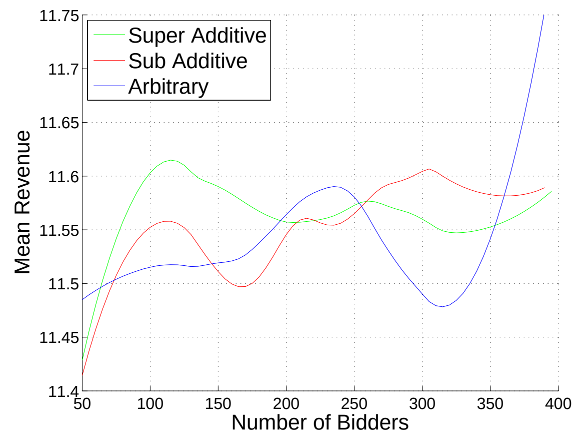

Figure 3. Log curve for Revenue of VCG-based mechanism

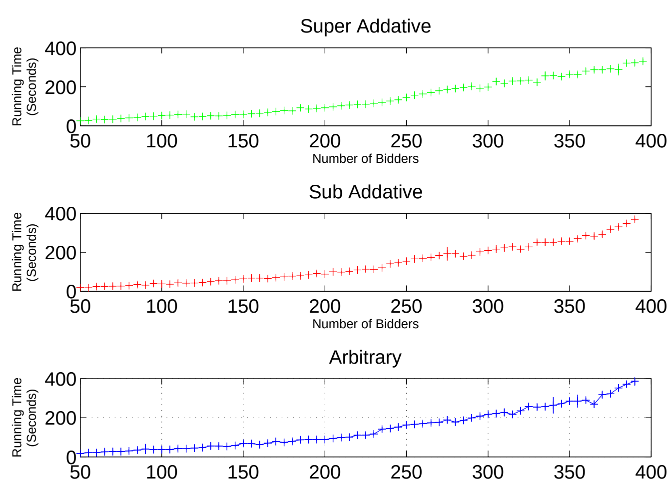

Figure 4. Running Time

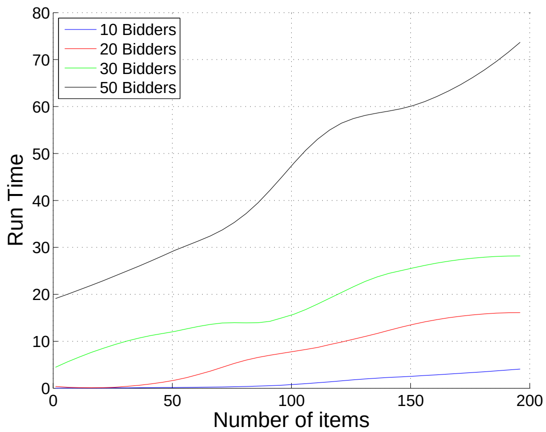

Figure 5. Run Time with varying bundles

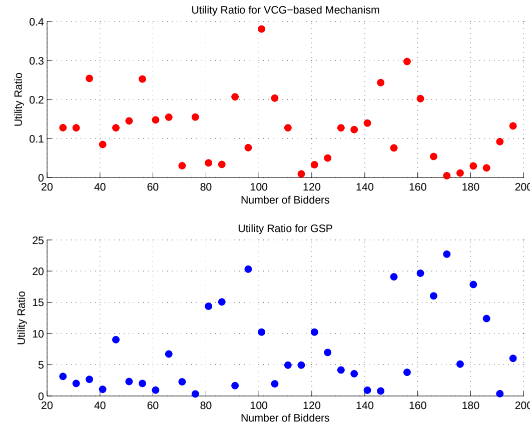

Figure 6. Utility Ratio

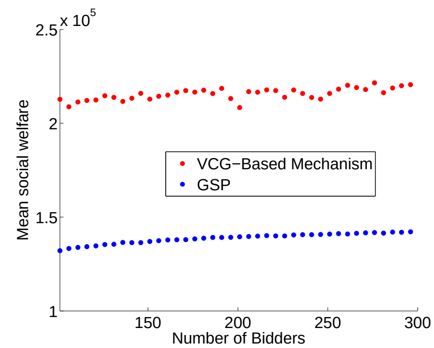

Figure 7. Social Welfare (Arbitrary Setting)

We also conducted experiments to evaluate the running time by fixing the number of bidders and increasing the number of bundles successively. We discovered that as the number of bundles increases, the running time increases almost linearly. With a higher number of bidders, the running time tends to increase faster. The same can be seen in Figure 5.

VII. VCG-BASED MECHANISM AND GSP MECHANISM

To compare the truthfulness, social-welfare and revenue of our VCG-based versus GSP, we conducted the experiments with same inputs for the both. However, since the GSP mechanism cannot handle bundles, it was not considered in its input.

A. Truthfulness

To quantify the truthfulness of our approach as explained in sub-section V-A, we conducted many experiments and took the average gain/loss in utility of a lying bidder. Although in some particular instances the lying bidder was able to increase her (his) utility in our mechanism, in overall multiple iterations, the lying bidder was not able to increase the average utility. At the same time, for GSP we observed that there were random increases in utility. In particular, most of the time the lying bidder was able to increase the utility because even if (s)he bids very high as compared to market valuation of the item, (s)he ended up winning the item and still paying the bid-value near to market price as quoted by other losing bidders. In GSP, the items are arranged in specific order and bid-values also have similar order. Hence, a lying bidder who was earlier winning a lower order item or no item at all, might win higher value item and gain substantially high utility by paying something close to market value as quoted by other bidders. In case of GSP, one single lying bidder can change the complete allocation pattern; this might account for the random gain in utility we see.

For the VCG-based mechanism, we observed that over many iterations, the lying bidder either got either zero or less utility for lying. Over a large number of experiments (about 1000 for a given number of players), we found that overall the lying bidder did not achieve substantially more than he could have achieved by telling the truth. In our truthfulness experiments, we used a grid size of 5×5, E-factor $= 1$ and S-factor $= 0.01$. In Figure 6, the curves shown in blue represent the results for the VCG-based mechanism and the ones in red represent those for the GSP mechanism. We can observe that the utility ratio for true valuation and lying valuations is less than one for our mechanism. Thus, overall, the lying bidder is not able to increase her (his) utility by lying. But, in case of GSP, the lying bidder is able to increase the utility by a substantial amount.

B. Social-Welfare

The magnitude of difference between the social-welfare of GSP and VCG depends on the parameters like E-factor and S-Factor. The results are presented in Figures 7, 8 and 9. In case of super-additive, our VCG-based mechanism is able to generate higher social-welfare as compared to GSP. This can be attributed to the higher economic capacity of combanitorial auction. The social-welfare in arbitrary setting is also observed to be higher than GSP. In case of sub-additive the social-welfare of VCG-based mechanism is lower than GSP. This can be attributed to the fact that sub-additive bids do not increase the overall valuation of bundle as a bid.

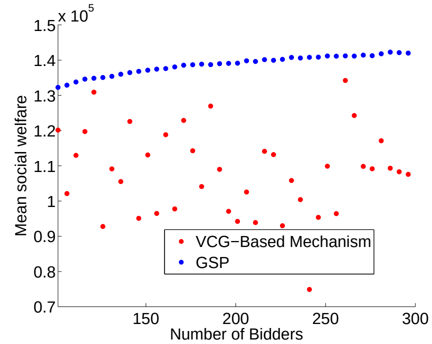

Figure 8. Social Welfare (Sub-Additive Setting)

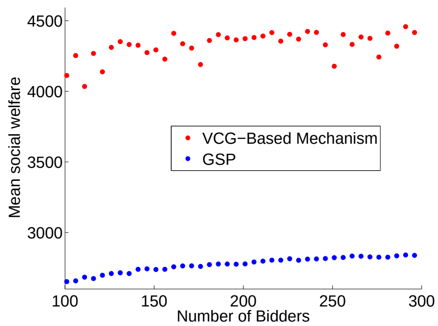

Figure 9. Social Welfare (Super-Additive Setting)

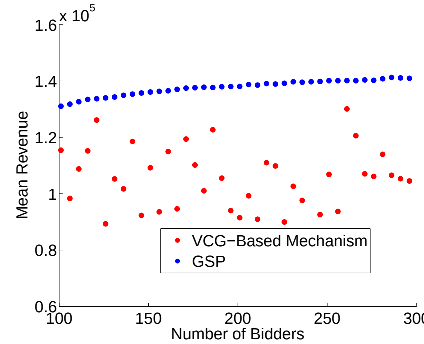

Figure 11. Revenue (Sub-Additive Setting)

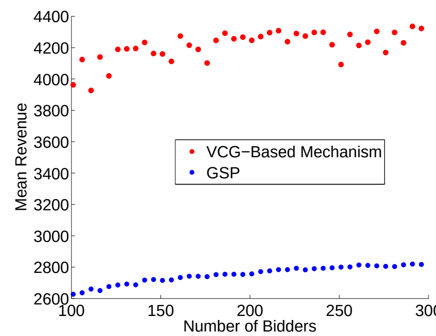

Figure 12. Revenue (Super Additive Setting)

C. Revenue

The revenue generated by our mechanism was higher than that of GSP in super-additive and arbitrary settings. It follows similar pattern as social-welfare. The magnitude of difference between the revenue of GSP and VCG-based mechanism depends on the E-factor and S-Factor. The results are presented in Figures 10, 11 and 12 for various settings.

VIII. CONCLUSION

In this paper we provided, for (what we believe to be) the first time, an implementation of a VCG-based mechanism for display Ad auctions. Our experiments show that this mechanism is more resilient to lying bidders as compared to GSP, and has reasonable time requirements for expected problem sizes. We also found out experimentally that the implemented mechanism can offer substantially better revenue and social welfare than GSP in many cases. One of the reasons for this is that the combinatorial setting allows for expressing valuations over bundles and generally, bundles have more economic values than single items.

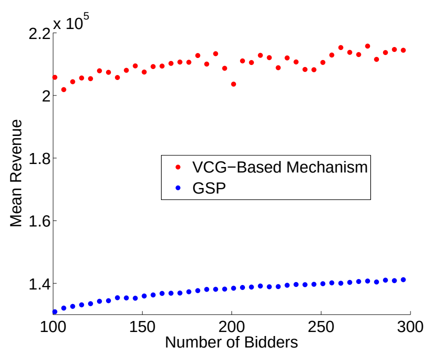

Figure 10. Revenue (Arbitrary Setting)

REFERENCES

[1] "Iab internet advertising revenue report 2012 full year results," http://www.iab.net/media/file/IAB_Internet_Advertising_Revenue_Report_FY_2012_rev.pdf, retrieved: 05, 2015.
[2] "Google ads display network," http://www.google.ae/ads/displaynetwork/, retrieved: 05, 2015.
[3] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Algorithmic Game Theory. Cambridge University Press, 2007.
[4] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," in FOCS, 2005, pp. 595–604.
[5] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," Journal of the ACM (JACM), vol. 58, no. 6, 2011, p. 25.
[6] G. Christodoulou, K. Elbassioni, and M. Fouz, "Truthful mechanisms for exhibitions," in WINE, 2010, pp. 170–181.
[7] M. Hoefer, T. Kesselheim, and B. Vöcking, "Approximation algorithms for secondary spectrum auctions," in SPAA, 2011, pp. 177–186.
[8] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords," American Economic Review, vol. 97, no. 1, 2007, pp. 242–259, retrieved: 05, 2015. [Online]. Available: http://www.aeaweb.org/articles.php?doi=10.1257/aer.97.1.242
[9] K. M. Elbassioni, K. Mehlhorn, and F. Ramezani, "Towards more practical linear programming-based techniques for algorithmic mechanism design," CoRR, vol. abs/1408.1577, 2014. [Online]. Available: http://arxiv.org/abs/1408.1577
[10] D. S. Evans, "The online advertising industry: Economics, evolution, and privacy," The journal of economic perspectives, 2009, pp. 37–60.

[11] Z. Wei and M. Lin, “Auction vs. posted-price: Market mechanism, lender behaviors, and transaction outcomes in online crowdfunding,” Posted-Price: Market Mechanism, Lender Behaviors, and Transaction Outcomes in Online Crowd-Funding (September 1, 2013), 2013.

[12] B. Edelman and M. Ostrovsky, “Strategic bidder behavior in sponsored search auctions,” Decision support systems, vol. 43, no. 1, 2007, pp. 192–198.

[13] K. Asdemir, “Bidding patterns in search engine auctions,” in Second Workshop on Sponsored Search Auctions. Citeseer, 2006.

[14] M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, A. R. Karlin, C. Mathieu, and M. Schwarz, “Greedy bidding strategies for keyword auctions,” in Proceedings of the 8th ACM conference on Electronic commerce. ACM, 2007, pp. 262–271.

[15] N. Nisan and A. Ronen, “Algorithmic mechanism design (extended abstract),” in Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing, ser. STOC ’99. New York, NY, USA: ACM, 1999, pp. 129–140. [Online]. Available: http://doi.acm.org/10.1145/301250.301287

[16] P. Cramton, Y. Shoham, and R. Steinberg, “Combinatorial auctions,” 2006, retrieved: 05, 2015.

[17] T. Sandholm, “Algorithm for optimal winner determination in combinatorial auctions,” Artificial Intelligence, vol. 135, no. 12, 2002, pp. 1 – 54. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000437020100159X

[18] K. Leyton-Brown, M. Pearson, and Y. Shoham, “Towards a universal test suite for combinatorial auction algorithms,” in Proceedings of the 2Nd ACM Conference on Electronic Commerce, ser. EC ’00. New York, NY, USA: ACM, 2000, pp. 66–76. [Online]. Available: http://doi.acm.org/10.1145/352871.352879

[19] S. Gujar and Y. Narahari, “Optimal multi-unit combinatorial auctions,” Operational Research, vol. 13, no. 1, 2013, pp. 27–46.

[20] A. Archer, C. Papadimitriou, K. Talwar, and É. Tardos, “An approximate truthful mechanism for combinatorial auctions with single parameter agents,” in SODA ’03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003, pp. 205–214.

[21] L. Ausubel and P. Milgrom, “The lovely but lonely vickrey auction,” 2006, p. 1740.

[22] S. Dughmi and T. Roughgarden, “Black-box randomized reductions in algorithmic mechanism design,” in Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on, 2010, pp. 775–784.

[23] M. Babaioff and L. Blumrosen, “Computationally-feasible truthful auctions for convex bundles,” in In RANDOM+APPROX, 2004, pp. 64–75.

[24] “IBM ILOG CPLEX Optimizer,” urlhttp://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, Last 2010.

[25] D. Bienstock and G. Iyengar, “Approximating fractional packings and coverings in o(1/epsilon) iterations,” SIAM J. Comput., vol. 35, no. 4, 2006, pp. 825–854.

[26] M. D. Grigoriadis and L. G. Khachiyan, “A sublinear-time randomized approximation algorithm for matrix games,” Operations Research Letters, vol. 18, no. 2, 1995, pp. 53 – 58.

[27] N. Garg and J. Könemann, “Faster and simpler algorithms for multicommodity flow and other fractional packing problems,” in 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1998, pp. 300–309.

[28] R. Khandekar, “Lagrangian relaxation based algorithms for convex programming problems,” Ph.D. dissertation, Indian Institute of Technology, Delhi, 2004.

[29] C. Koufogiannakis and N. E. Young, “Beating simplex for fractional packing and covering linear programs,” in 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2007, pp. 494–504.

[30] S. A. Plotkin, D. B. Shmoys, and É. Tardos, “Fast approximation algorithms for fractional packing and covering problems,” in FOCS, 1991, pp. 495–504.

[31] N. E. Young, “Sequential and parallel algorithms for mixed packing and covering,” in FOCS, 2001, pp. 538–546.

[32] N. Nisan, “Bidding and allocation in combinatorial auctions,” in Proceedings of the 2nd ACM conference on Electronic commerce. ACM, 2000, pp. 1–12.

[33] C. Gallo, “A data set generation algorithm in combinatorial auctions,” in Computer as a Tool, 2005. EUROCON 2005.The International Conference on, vol. 1, Nov 2005, pp. 744–747.

[34] C. Boutilier, M. Goldszmidt, and B. Sabata, “Sequential auctions for the allocation of resources with complementarities,” in Proceedings of the 16th International Joint Conference on Artifical Intelligence - Volume 1, ser. IJCAI’99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 527–534. [Online]. Available: http://dl.acm.org/citation.cfm?id=1624218.1624294