Computational Bundling for Auctions

Christian Kroer
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA, USA
ckroer@cs.cmu.edu

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA, USA
sandholm@cs.cmu.edu

ABSTRACT

Revenue maximization in combinatorial auctions (and other multidimensional selling settings) is one of the most important and elusive problems in mechanism design. The optimal design is unknown, and is known to include features that are not acceptable in many applications, such as favoring some bidders over others and randomization. In this paper, we instead study a common revenue-enhancement approach—bundling—in the context of the most commonly studied combinatorial auction mechanism, the Vickrey-Clarke-Groves (VCG) mechanism. A second challenge in mechanism design for combinatorial auctions is that the prior distribution on each bidder's valuation can be doubly exponential. Such priors do not exist in most applications. Rather, in many applications (such as premium display advertising markets), there is essentially a point prior, which may not be accurate. We adopt the point prior model, and prove robustness to inaccuracy in the prior. Then, we present a branch-and-bound framework for finding the optimal bundling. We introduce several techniques for branching, upper bounding, lower bounding, and lazy bounding. Experiments on CATS distributions validate the approach and show that our techniques dramatically improve scalability over a leading general-purpose MIP solver.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; J.4.a [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms; Theory; Economics

Keywords

Bundling; search; integer programming; combinatorial auctions; VCG; revenue maximization

1. INTRODUCTION

Revenue maximization in combinatorial auctions (and other multidimensional selling settings) is one of the most important and most elusive problems in mechanism design. The optimal auction for a single item is known [32] and has been generalized to multiple units of one item [30], but the problem remains open even with just two items.[1] The fact that the general optimal combinatorial auction mechanism is unknown is not a coincidence: even a special case of the problem of designing the highest-revenue deterministic multi-item auction is NP-complete, even in the private values setting [15]. This suggests that, unlike for single-item settings, a concise characterization of optimal combinatorial auctions cannot exist. This is one of the key motivations for *automated mechanism design* where an algorithm is used to design the mechanism for the setting (prior probability distribution) at hand (e.g., [13, 14, 15, 37, 27, 28, 10]).

Even with automated design, and putting aside the computational complexity of the design problem, it is not clear that optimal combinatorial auctions are viable in practice, for the following reasons: (**a**) The revenue-optimal mechanism includes features that are not acceptable in many applications, such as favoring some bidders over others, and randomization. (**b**) The optimal mechanism is difficult to understand. This, itself, can be a deterrent to its adoption. (**c**) Even in the private values setting, the prior distribution on each bidder's valuation can have support the size of which is doubly exponential. Specifically, if there are m items and a bidder can have any of k values for each bundle, the support of the prior has k^{2^m-1} points because that is the size of the bidder's type space. Such prior distributions have not been (and cannot be) constructed in most applications. In this paper, we study a practical automated mechanism design setting where we avert these problems.

We avert the first two problems by only considering one common, practical way of increasing revenue, *bundling* items [33, 11, 24, 18, 6, 25, 9, 22, 45]. Our mechanisms will be fair in the sense that they are symmetric across bidders, and deterministic, unlike the optimal auction. Specifically, we will develop algorithms for optimal bundling in the context of the most commonly studied combinatorial auction mechanism, the *Vickrey-Clarke-Groves mechanism (VCG)* [43, 12, 19].[2] In the VCG—even with bundling—each bidder's dominant strategy is to bid truthfully.

We avert the third problem by not assuming that we have access to such a prior. In many (arguably most) applications there is essentially just a point prior, which may not be accurate. In other words, the seller has expectations about how much bidders would be willing to pay for various bundles, but the seller does not have a sophisticated probabilistic model about any bidder's valuations (which are not independent across bundles). This is the case, for example, in TV advertising sales and in premium, guaranteed display advertising sales. (In both of those markets, the sales occur manually, and the inventory is implicitly bundled in ad hoc ways today, with typically 2-4 targeting attributes.

[1]Relatively simple special cases have been solved, yielding interesting insights (e.g., [3, 5]).

[2]While the VCG itself has taken some time to get adopted in applications, it has been successfully adopted, for example, by Facebook for its advertising auctions.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

Another, very practically oriented, potential application of our work is to help an auction designer choose bundles through exploratory search. If no distribution is available, the auction designer could generate sets of reasonable bids in an iterative fashion, compute optimal bundlings, and choose a final bundling based on the intuitions gained from this.

Therefore, we adopt the point prior model. However, we also acknowledge the fact that in practice the point prior might not be accurate; we prove robustness of our approach to inaccuracy in the prior. One might also ask why we do not simply make take-it-or-leave-it offers to take the entire surplus for the seller given that we have a point prior. The reason we do not use that mechanism is that it is highly nonrobust to error in the prior: even slight overpricing on a bundle will cause the revenue from that bundle to drop to zero.

We develop a custom branch-and-bound framework for finding the optimal bundling. We show that algorithms in that framework scale significantly better than a leading general-purpose integer program solver, CPLEX. We design and compare several techniques for upper bounding, lower bounding, branching, and lazy child node evaluation. Experiments on the leading combinatorial auction test suite, CATS [26], validate the approach.

Our approach to bundling is computational. The goal is not to build insight from manual analysis that can then loosely be applied to practice—although the computational approach can help develop insight as well and fuel future theory. Rather, the goal is to develop a computational methodology that can be used in practice. It is therefore key that we make the computational techniques scalable.

1.1 Related work

Most prior work on bundling has been in the context of posted prices, for example, catalog pricing. Bundle pricing in economics has often focused on analyzing two-item settings [41, 1, 16, 20, 31, 40]. (One exception is that Armstrong [2] examines m-item settings, but places severe restrictions on buyers' utility functions. Another exception is that Manelli et al. [29] provide results for when bundled catalog sales are optimal, mainly in the two- and three-item settings.) There has been some computational work on bundle pricing, for example, mixed integer programming for optimizing bundle prices [21] and data-driven pricing of car configurations [35].

Benisch et al. [7] present a framework for automatically suggesting high-profit bundle discounts based on historical purchase data. There has also been work on pricing bundles of information goods, where it is usually assumed that valuations are based only on the size of the bundle and there are no marginal costs [25, 9, 6]. The operations research literature has also addressed the information goods setting. For example, Hitt et al. [22] and Wu et al. [45] consider a bundle pricing mechanism for information goods that allows customers to choose up to m' items from a larger pool of m items.

Walsh et al. [44] study inventory bundling in premium display advertisement campaign selling, with the goal of making the winner determination problem smaller and easier. Others have recently studied that problem as well [42, 23]. That objective is different than revenue maximization.

There has been some work on bundling in auctions. Several papers considered the problem of bundling all items versus separate sales [33, 11]. Armstrong [3] shows that in two-item auctions with two valuations per bidder, the revenue-maximizing auction is efficient. Avery et al. [5] show that this does not hold in general two-item settings, and that randomization can increase revenue.

Prior work on *automated mechanism design* has produced deterministic and randomized mechanisms for revenue-maximizing combinatorial auctions [14, 15, 27, 28]. Since our paper focuses on the VCG mechanism, our bundling is deterministic. There has been work on generalizing the VCG to higher-revenue dominant-strategy auction mechanisms. Likhodedov et al. [27, 28] study a generalization of VCG called *virtual valuations combinatorial auctions (VVCAs)* and a generalization of them, *affine maximizer combinatorial auctions*, which are auctions where the VCG is run on affine transformations of the bids. Those mechanism design algorithms scaled to a handful of items. For the restricted setting of additive valuations, Jehiel et al. [24] analyze a subclass of VVCAs called λ-*auctions*, which are nevertheless rich enough to allow the bundling to depend on the bids. Our approach of bundling first and then simply running VCG is easier for buyers and sellers to understand. Also, none of the prior papers present bundling algorithms. Core-selecting combinatorial auctions are another form of combinatorial auctions that has recently become popular in spectrum auctions [17]. They can generate higher revenue than the VCG. However, equilibria are not known (except in the known-valuations setting), so game-theoretic simulation of revenue properties is not possible. Core-selecting combinatorial auctions can also suffer from revenue deficiencies that bundling can alleviate.

To our knowledge, there has been no prior work on our setting.

2. PROBLEM FORMULATION

We assume that we have a set of bidders $N = \{1, \ldots, n\}$, a set of items $M = \{1, \ldots, m\}$, and a set of bids B. (The bids represent the, possibly inaccurate, point prior, and are given before actual bids are received.) B_i is the set of all bids for bidder i, and B_{-i} the set of all bids that do not belong to i. A bid is a tuple $\langle S_j, v_j\rangle$, where $S_j \subseteq M$ is the set of items that the bid wants and $v_j \geq 0$ is a valuation. For ease of exposition, we sometimes use $[k]$ to refer to the set $\{1, \ldots, k\}$ for $k \in \mathbb{N}$.

We denote an allocation of items to bidders by α, and α_i is the set of items allocated to bidder i in the allocation. We overload $v_i(\alpha)$ to be bidder i's valuation for α_i.

We work in the standard combinatorial auction setting with the XOR bidding language [36] so each bidder can have at most one of her bids win. Every item can be assigned to at most one bidder. There are *no externalities*: the valuation for each bidder i depends only on the items that i receives. There is *free disposal*: the value of a subset of items $M' \subseteq M$ for bidder i is less than (or equal to) the value of M. A bundle $b \subseteq M$ is a set of items from M. A *bundling* ϕ is a set of bundles that partitions M. We denote the set of all possible bundlings by Φ.

We say that two bids i, j *intersect* if their item sets overlap: $S_i \cap S_j \neq \emptyset$. Bundling can introduce additional overlap between bids, and for this reason we need a more general notion of overlap. Two bids i, j *bundling-intersect* in a given bundling ϕ if they intersect or there exists at least one item in each of the two bids so that those two items have been included in the same bundle. Formally, two bids i, j bundling-intersect in bundling ϕ if they intersect or $\exists a_i \in S_i, a_j \in S_j, b \in \phi$ such that $a_i \in b, a_j \in b$. We denote the set of bundling-intersecting items of the two sets by $S_i \sqcap S_j$.

We will be running the VCG to auction the bundles simultaneously. In the vanilla VCG without bundling, the allocation of items to bidders is computed so that it maximizes social welfare (henceforth referred to as welfare): $\alpha^* = \arg\max_\alpha \sum_{i=1}^n v_i(\alpha)$. The payment from each bidder i is $p_i = W^{*,-i}_{M,B} - W^{*,-i}_{M_{-i},B}$, where $W^{*,-i}_{M,B}$ is the optimal welfare where i receives no items, and we let $W^{*,-i}_{M_{-i},B} = \sum_{j \neq i} v_j(\alpha^*)$ be the welfare of the other agents in α^*. We denote by $W_B(\alpha)$ the welfare of an allocation α for bids B.

In our setting, we have to take the bundling into account in the VCG. For a given bundling $\phi \in \Phi$, the VCG allocation α^* is computed as before, but with the added constraint that no two items that are in the same bundle can be allocated to different bidders (i.e., for any two winning bids i and j, $S_i \sqcap S_j = \emptyset$). We denote the welfare of such an allocation by $W^*_{\phi,B} = \max_{\alpha \in A_\phi} \sum_{i \in N} v_i(\alpha)$, and the welfare of the welfare-maximizing allocation over the bundling, where bidder i is excluded, as $W^{*,-i}_{\phi,B} = \max_{\alpha \in A_\phi} \sum_{j \neq i} v_j(\alpha)$. Similarly, running VCG on bundling ϕ yields payments

$$p_i = W^{*,-i}_{\phi,B} - W^{*,-i}_{\phi_{-i},B}$$

from each bidder i. Here, ϕ_{-i} is the set of bundles from ϕ that are not allocated to bidder i. The goal in optimal bundling is to find a bundling ϕ^* such that the revenue $r_{\phi^*} = \sum_{i=1}^n p_i$ is maximized.

3. BASIC PROPERTIES OF THE SETTING

In this section we study some important basic properties of the proposed approach.

Number of bundlings The number of bundlings grows extremely rapidly with the number of items. The number of ways to bundle (exhaustively partition) m items is called the Bell number, $\mathcal{B}_m$. The Bell numbers can be defined recursively: $\mathcal{B}_m = \sum_{k=0}^{m-1} \binom{m-1}{k} \mathcal{B}_k$. Sandholm et al. [38] proved that $\mathcal{B}_m \in \omega(m^{\frac{m}{2}})$. Berend et al. [8] proved that $\mathcal{B}_m < (\frac{0.792m}{\ln(m+1)})^m$.

NP-hardness We first prove hardness of our problem (revenue-maximizing bundling in the VCG setting with a point prior).

THEOREM 1. *Finding the optimal bundling is NP-hard.*

PROOF. The proof is by reduction from bin packing. The bin packing problem is the following: Given a set of n objects of sizes $x_1, ..., x_n$, and positive integers k and V, is it possible to fit the n objects into k bins of size V?

For each bin $i = 1, ..., k$, we generate $n+1$ items: a_i^{bin} and $G_i = \{g_1^i, ..., g_n^i\}$, and two bidders with bids $b_i^{bin}, b_{i'}^{bin}$ respectively, where $S_i^{bin} = \{a_i^{bin}\} \cup G_i, S_{i'}^{bin} = \{a_i^{bin}\}$, and valuations $v_i^{bin} = V + M, v_{i'}^{bin} = M$, for some $M > V, M > \sum_{j=1}^n x_j$. The set $G_i = \{g_1^i, ..., g_n^i\}$ is there to allow up to n other bids to intersect with b_i^{bin}, without having the bids themself intersecting. For each of the objects $j = 1, ..., n$, we generate an item a_j^{obj} and a bidder with bid b_j^{obj} with $S_j^{obj} = \{a_j^{obj}\}$, and valuation $v_j^{obj} = x_j$. Bundling an object item a_j^{obj} with some bin item g_i^l represents putting object j in bin i. The only pairs of bids that overlap before bundling are $b_i^{bin}, b_{i'}^{bin}$.

We claim that there is a solution to the bin packing problem iff there is some bundling with revenue $r = kM + \sum_{j=1}^n x_j$ for the above bundling problem. Now, assume that we have some solution with revenue $r \geq kM + \sum_{j=1}^n x_j$. Clearly, it cannot be the case that any two items such that $S_i^{bin} \cap S_j^{bin} \neq \emptyset$ for some bids b_i^{bin}, b_j^{bin} are bundled together, as this would cause revenue $r \leq (k-1)M + \sum_{j=1}^n x_j < kM$, since only $k-1$ bids with valuation above or equal to M can now win, and the remaining bids can have aggregate valuation at most $\sum_{j=1}^n x_j < M$. Hence, we are guaranteed kM revenue from the bids $\{b_i^{bin}, b_{i'}^{bin} : i = 1, ..., k\}$, and we know that they will not be bundled together. Now, each of the bids b_j^{obj} must be contributing x_j to revenue (either by paying that amount or causing one other bidder to pay that much more), since they are the only bids left that can be made bundling-intersecting with any other bid through bundling, and they have no competition if unbundled. We also know that any two bids b_i^{obj}, b_j^{obj} cannot be made bundling-intersecting. If they were, the revenue obtained from the two would be at most $\max(v_i^{obj}, v_j^{obj})$ (since they can be made bundling-intersecting with at most one winning bid, and alternatively if one of the two bids wins, its payment cannot exceed its valuation). It follows that the bids b_j^{obj} must bundling-intersect with the bids b_i^{bin}, which are already contributing kM revenue. Let B_i^{obj} be the set of bids b_j^{obj} that bundling-intersect with b_i^{bin}. We must have $M + \sum_{j \in B_i^{obj}} x_j \leq M + V$, since otherwise $r < kM + \sum_{j=1}^n x_j$, as some x_j is then not contributing its full valuation. Now, we can take the solution and turn it into a solution to the bin packing problem. Since we know that all b_j^{obj} bundling-intersect with one b_i^{bin} each, we take each such j and assign that object to bin i. Since $\sum_{j \in B_i^{obj}} x_j \leq V$ for all i, we know that this is a valid packing.

Conversely, if there is a solution to the bin packing problem, there is a bundling such that all bids b_j^{obj} bundling-intersect with one b_i^{bin} each, and $M + \sum_{j \in B_i^{obj}} v_j^{obj} = M + \sum_{j \in B_i^{obj}} x_j \leq V + M$ for all i. Hence we can get revenue at least $kM + \sum_{j=1}^n x_j$. □

In the instances generated by the reduction in the proof of Theorem 1, the winner determination problem is easy. This shows that the bundling problem is hard in itself. The NP-completeness of winner determination does, however, imply that the general optimal bundling problem is not contained in NP. Given a bundling with proposed revenue k, verification that it achieves revenue k requires computing the (exact, due to revenue nonmonotonicity) social-welfare maximizing outcomes, in order to compute payments.

Ghosh et al. [18] similarly show NP-hardness for the problem of computing the optimal bundling in their model, which assumes additive valuations. Their NP-hardness result relies on this, as their reduction uses the fact that bundle valuations can be expressed compactly in this setting. For the XOR language, representing the same valuations would take an exponential number of constraints, and thus hardness of our setting does not follow from theirs.

Revenue is nonmonotonic in bidders It is known that the VCG is not revenue monotonic in bidders [4], that is, adding bidders can decrease revenue. Rastegari et al. [34] characterize a broader class of mechanisms for which revenue monotonicity cannot be achieved. Our setting falls outside of this class of mechanism. Nevertheless, we can show revenue nonmonotonicity in our setting. Consider the valuations in Figure 1 Left. The optimal bundling is to sell the items separately, in which case bidders 3 and 4 receive items Y and X respectively, with payments 3 and 5, yielding total revenue of 8. If we remove bidder 4, the optimal bundling is still to sell the items separately, where bidder 1 wins both items, with a payment of 9, which is also the revenue. Thus removing bidder 4 can increase revenue.

	X	Y	XY
Bidder 1	0	0	10
Bidder 2	4	0	4
Bidder 3	0	5	5
Bidder 4	7	0	7

	X	Y	XY
Bidder 1	10	0	10
Bidder 2	5	0	5
Bidder 3	0	2	2
Bidder 4	0	1	1
Bidder 5	0	20	20

Figure 1: Left: Bidder valuations for a game with 4 bidders and 2 items. Right: Bidder valuations for a game with 5 bidders and 2 items. In both tables, the XY column denotes bidder valuations for the bundle.

Coarseness of the optimal bundling is nonmonotonic in bidders The optimal bundling can become coarser with the addition of a bidder. An example is given in Figure 1 Right. The optimal bundling without Bidder 5 is to sell X and Y separately, whereas with 5 it is optimal to bundle.

Low worst-case revenue It is well known that the VCG can be arbitrarily far from optimal in terms of revenue. Consider two bidders, and items X and Y, where bidder 1 bids v_a for X and bidder 2 bids v_b for Y, and there are no other bids. In this case the VCG revenue is 0. For settings such as this, optimal bundling can have arbitrarily high revenue lift over VCG since we can bundle X and Y, and earn $\min(v_a, v_b)$.

Even with optimal bundling in the VCG, we can have arbitrarily high loss compared to what is possible. This is because the seller would give away $|v_1 - v_2|$ of the surplus (and this part can be arbitrarily large) while with reserve prices of v_1, v_2 on the items, respectively, would allow the seller to capture the entire surplus.

Robustness of the proposed approach Another practical revenue-enhancement technique is reserve pricing. Since a point prior is typically imprecise in practice, using reserve pricing can lead to high revenue loss. Consider again the example in th previous paragraph, but where the point prior is that each bidder i has valuation $v_i = v$ and the bidder's true distribution p is $p(v_i = v) = \epsilon, p(v_i = v - \delta) = 1 - \epsilon$. In this case, if the reserve price is set between $v - \delta$ and v, the expected loss in revenue is $2(1 - \epsilon)v$. In contrast to this, optimal bundling in the VCG would only have an expected loss of $2\epsilon v + (1 - 2\epsilon)(v - \delta) < v$. As shown in the example above, optimal bundling is more robust to inaccuracy in the point-prior than reserve-pricing. This is important, as the availability of precise point-priors is unlikely in real-world applications. In this section we make this notion of robustness more precise, as we show the following result. In contrast, bundling for the VCG is robust to error in the (point) prior:

THEOREM 2. *The revenue from the VCG with optimal bundling (which may change based on bidder valuations) is Lipschitz continuous in the valuations of the bidders with Lipschitz constant $n - 1$. This bound is tight.*

To prove this, we first prove the following lemma.

LEMMA 3. *The revenue from the VCG with any fixed bundling is Lipschitz continuous in the valuations of the bidders with Lipschitz constant $n - 1$.*

PROOF. The revenue obtained from VCG on a bundling ϕ can be written as

$$r_\phi = -(n-1) \cdot W^*_{\phi,B} + \sum_{i=1}^{n} W^{*,-i}_{\phi,B} \tag{1}$$

where $W^*_{\phi,B}$ is the welfare of the welfare-maximizing allocation for the bids B, and $W^{*,-i}_{\phi,B}$ denotes the same when bidder i's bids are excluded. Thus, it suffices to show that all the terms in Equation 1 are Lipschitz continuous in bidder valuations.

Assume that some bidder i changes his valuation for some bid j, and let B' be the new set of bids, where v'_j is the new valuation for bid j. We now show that the change in welfare $\Delta W = |W^*_{\phi,B} - W^*_{\phi,B'}|$ is bounded by $\Delta v_j = |v_j - v'_i|$. For any allocation α such that j is winning, we get that the change in welfare is Δv_j, whereas for any allocation α' such that j is not winning, the welfare remains the same. Hence, the previously winning allocation α can increase or decrease by at most Δv_j. Since $W_B(\alpha) \geq W_B(\alpha')$ for all α', the new winning allocation α^* satisfies $W_{B'}(\alpha^*) \leq W_B(\alpha) + \Delta v_j$. For the lower bound, we have $W_{B'}(\alpha) \geq W_B(\alpha) - \Delta v_j$, and hence α^* must satisfy $W_{B'}(\alpha^*) \geq W_B(\alpha) - \Delta v_j$.

Since all terms are welfare maximizations over different sets of bidders, this proves that r_ϕ is Lipshitz continuous in bidder valuations, with a Lipschitz constant of $n - 1$ (because the $n - 1$ term and n terms in the summation change in opposite directions, and the summation over n terms only has $n - 1$ terms that can change, as $W^{*,-i}_{\phi,B}$ does not depend on v_j). $\square$

With this lemma, we are now ready to prove Theorem 2.

PROOF. There are two possible cases. In the first case, the optimal bundling does not change, and in the second case it changes. The proof of the first case is immediate from Lemma 3. For the second case, let ϕ_1 and ϕ_2 be the old and new optimal bundlings, respectively. By Lemma 3 we can bound the revenue of a bundling under the new valuation using the old valuation:

$$r_B(\phi) - (n-1) \cdot \Delta v_i \leq r_{B'}(\phi) \leq r_B(\phi) + (n-1) \cdot \Delta v_i$$

By optimality,

$$r_{B'}(\phi_2) \geq r_{B'}(\phi_1) \geq r_B(\phi_1) - (n-1) \cdot \Delta v_i$$

From the fact that ϕ_1 is optimal for the original bids (B) we know that $r_B(\phi_2) \leq r_B(\phi_1)$ and hence

$$r_B(\phi_2) + (n-1) \cdot \Delta v_i \leq r_B(\phi_1) + (n-1) \cdot \Delta v_i$$

By Lemma 3, the left hand side of this inequality is an upper bound for $r_{B'}(\phi_2)$. Thus we get

$$r_B(\phi_1) - (n-1) \cdot \Delta v_i \leq r_{B'}(\phi_2) \leq r_B(\phi_1) + (n-1) \cdot \Delta B_i$$

So, the new optimal revenue is bounded both above and below as shown above, and is (by the formulas above) Lipschitz continuous with Lipschitz constant $n - 1$.

Finally, we show that the bound is tight. Consider the case with 2 items X,Y and 3 bidders $\{b_1, b_2, b_3\}$, where b_1 has valuation 1 for X, b_2 has valuation 1 for Y and b_3 has valuation $\frac{1}{2}$ for X and for Y. The optimal bundling yields revenue 1, but if b_3 increases his valuation for both items to 1, the revenue increases to 2, and the increase in revenue is $1 = \frac{1}{2} \cdot (n-1)$. $\square$

This shows that our approach is robust to inaccuracy in the prior in the following sense: If the point-prior turns out to be inaccurate, the change in revenue is at worst linear in the sum of inaccuracies in the prior. It is further worth pointing out that for m items, at most $3 \cdot m^2$ bids can affect the payments. Thus, the number of bids potentially being exponential in the number of items does not cause huge revenue shifts, even if there is inaccuracy for all bids.

4. MIXED INTEGER PROGRAM (MIP)

One approach to finding the revenue-maximizing bundling is to formulate the problem as a mixed integer program (MIP), and then use a general-purpose MIP solver—such as CPLEX—to solve the formulation. In this section we develop such a formulation.

Figure 2 shows the formulation. The basic idea behind this MIP is that we have m potential bundles, and the boolean variables $\delta_{a,b}$ denote whether item a is assigned to bundle b. Based on these assignments, the VCG payments are computed. To break symmetries, we only allow each item $a = 1, ..., m$ to be assigned to bundles $\{1, ..., a\}$. Furthermore, items with index $a > b$ can only be assigned to the bundle with index b if the item with index b is also assigned to the bundle. Each bid j has boolean variables I_j and I^{-i}_j that denote whether the bid wins in the optimal allocation and optimal allocation excluding bidder i, respectively. Each bidder has boolean variables $\Pi_j(b)$ and $\Pi^{-i}_j(b)$ that denote whether the bidder is allocated bundle b in the respective allocations, and boolean variables $\pi_j(a, b)$ and $\pi^{-i}_j(a, b)$ that denote whether the bidder is allocated item a through bundle b in the respective allocations. Finally, each real-valued variable p_i denotes the payment that bidder i must make. The objective function, (2), is the sum over the payment variables of the bidders. Constraint 3 sets the payment for bidder i equal to the externality she imposes on the other bidders, i.e., her VCG payment. Constraints 4-5 ensure that a bid j can only be winning if that bidder is assigned all the items in S_j for each allocation. Constraints 6-7 ensure that each bundle is assigned to only one bidder in each allocation. Constraints 8-11 ensure that a bidder can only receive item a through bundle b in each allocation if the bidder wins the bundle, and $\delta_{a,b} = 1$, i.e., the item is in the bundle. Constraints 12-13 ensure that each bidder wins only one item in each allocation. Constraints 14-15 ensure that each item is assigned to only one bundle and they break symmetries. Finally, Constraint 16 ensures that the welfare-maximizing allocation is chosen for each bundling ϕ by ensuring that if all the $\delta_{a,b}$ that are necessary to achieve ϕ are active, then the winning bids are active.

$$\max \sum_{i=1}^{n} p_i \tag{2}$$

$$p_i \le \sum_{j \in B_{-i}} v_j \cdot I_j^{-i} - \sum_{j \in B_{-i}} v_j \cdot I_j \quad \forall i \in N \tag{3}$$

$$I_j \le \sum_{b=1}^{m} \pi_i(a,b) \quad \forall i \in N, j \in B_i, a \in S_j \tag{4}$$

$$I_j^{-i} \le \sum_{b=1}^{m} \pi_k^{-i}(a,b) \quad \forall i,k \in N, j \in B_k, a \in S_j \tag{5}$$

$$\sum_{i=1}^{n} \Pi_i(b) \le 1 \quad \forall b \in [m] \tag{6}$$

$$\sum_{k=1}^{n} \Pi_k^{-i}(b) \le 1 \quad \forall i \in N, b \in [m] \tag{7}$$

$$\pi_i(a,b) \le \Pi_i(b) \quad \forall i \in N, a \in [m], b \in [m] \tag{8}$$

$$\pi_k^{-i}(a,b) \le \Pi_k^{-i}(b) \quad \forall i,k \in N, a \in [m], b \in [m] \tag{9}$$

$$\pi_i(a,b) \le \delta_{a,b} \quad \forall i \in N, a \in [m], b \in [m] \tag{10}$$

$$\pi_k^{-i}(a,b) \le \delta_{a,b} \quad \forall i,k \in N, a \in [m], b \in [m] \tag{11}$$

$$\sum_{j \in B_i} I_j \le 1 \quad \forall i \in N \tag{12}$$

$$\sum_{j \in B_k} I_j^{-i} \le 1 \quad \forall i,k \in N \tag{13}$$

$$\sum_{b=1}^{a} \delta_{a,b} = 1 \quad \forall a = 1,...,m \tag{14}$$

$$\delta_{a,b} \le \delta_{b,b} \quad \forall b = 1,...,m, a = b,...,m \tag{15}$$

$$\sum_{(a,b) \in \alpha_\phi^*} \delta_{a,b} - I_j \le |\alpha_\phi^*| - 1 \quad \forall \phi \in \Phi, j \in B_{win(M')} \tag{16}$$

$$I_j, I_j^{-i}, \pi_j(a,b), \pi_j^{-i}(a,b) \in \{0,1\} \tag{17}$$

$$\Pi_j(a,b), \Pi_j^{-i}(a,b), \delta_{a,b} \in \{0,1\}, p_i \ge 0 \tag{18}$$

Figure 2: Mixed integer program for finding the optimal bundling.

The MIP model suffers from several limitations. First, the MIP has $\Omega(n|B|m^2)$ boolean variables, which rapidly becomes unmanageable. Second, and more importantly, Constraint 16 is required for every possible bundling, of which there are an extremely large number, as mentioned in Section 3; additionally, but less significantly, to generate each of these constraints, the welfare-maximizing allocation must be found, which is NP-hard in itself. This could potentially be alleviated by using constraint generation techniques but even this is unlikely to yield acceptable scalability, as each added constraint only cuts off solutions at that specific bundling, and nowhere else. In addition, this would require resolving the already large MIP every time a constraint is added.

5. CUSTOM BRANCH-AND-BOUND

We will now move on to discussing our custom branch-and-bound approaches. Later we show that these scale significantly better than general purpose MIP solving.

5.1 Branching scheme

To find the optimal bundling we introduce a custom branch-and-bound algorithm, FIND-BUNDLING. It is a tree search algorithm that branches on items. At each node in the search tree, the algorithm branches on an item, with each branch adding the item to a different bundle. One of the branches corresponds to adding it to the empty bundle. The algorithm explores nodes in best-first order. The revenue obtained from the best solution found so far is a global variable f^*; initially $f^* = 0$. The pseudocode is given in Algorithm 1.

ALGORITHM 1: FIND-BUNDLING

Input: Set of items M, Set of bids B
Output: Optimal revenue LB, Optimal bundling
1 LB$\leftarrow 0$
2 insert $(\{\{$BRANCHINGITEM$(M, B)\}\}, \infty)$ in OPEN
// Open is a priority queue of search tree fringe nodes,
// sorted in descending order of the second argument
3 **while** OPEN *not empty* **do**
4 (CURRENT, VAL) $\leftarrow$ next in OPEN
5 **if** UB(CURRENT,M, B)>LB **then**
6 i $\leftarrow$ BRANCHINGITEM(CURRENT,M, B)
7 **for** b *in* CURRENT **do**
8 CHILD $\leftarrow$ CURRENT with i added to b
9 f^* = MAX(LB(CHILD,M, B),f^*)
10 insert (CHILD, UB(CHILD,M, B)) in OPEN
11 CHILD $\leftarrow$ CURRENT with $\{i\}$ appended to the list of bundles
12 insert (CHILD, UB(CHILD,M, B)) in OPEN

FIND-BUNDLING starts out with a bundling consisting of a single item, chosen by the function BRANCHINGITEM(M,B) (Step 2). At each node, the next item i to branch on is chosen (Step 6), and in Step 7-10 FIND-BUNDLING creates a branch for each of the existing bundles, with i added to that bundle. For the last branch in Step 11-12, a new bundle is added with i as the lone item in that bundle. The branching factor at a given node is therefore the number of bundles already created plus one. For each node, the upper bound is used for deciding where in the ordered OPEN list the node is inserted.

5.2 Lower bounding

In Step 9 FIND-BUNDLING computes a lower bound at the node. If a high lower bound is found, we can update f^*, and thereby achieve better pruning.

We use the following technique for lower bounding. For any node v in the search tree, we simulate a VCG auction on the bundles decided on the search path from the root to v along with all the yet-undecided (i.e., yet unbundled) items. Our lower bound is then the sum of the VCG payments.

PROPOSITION 4. *The sum of the VCG payments from selling unbranched items separately is a valid lower bound.*

PROOF. One option for FIND-BUNDLING is to take the branch where every item is added in its own separate bundle for all yet-undecided items. This path yields exactly the auction that is used in the lower bound definition. □

In the rest of the paper, whenever we refer to the revenue of a node, we mean the value defined in this section.

To compute the lower bound, we make $n + 1$ calls (one overall and one with each bidder removed in turn) to a subroutine that does (optimal) combinatorial auction winner determination. We call that routine DETERMINE-WINNERS. We use the standard MIP formulation [36]. For the branch where the item is added alone in a bundle, we can reuse the bound from the parent node, as this is exactly the same MIP.

Typically in tree search/integer programming, if one does not use a lower-bounding technique for the yet-undecided variables, one simply (implicitly) uses a lower bound equal to the value from the variables that have been decided on the path from the root to the current node. For example, in winner determination for combinatorial auctions, one uses the sum of the values of the bids that have been accepted on that path (e.g., [36, 39]). Interestingly, in the bundling setting one needs to be more careful. For example, using just the bids that are only interested in items that have already been bundled on the path would not give a valid lower bound. The reason is that this could discard a bidder that causes revenue non-monotonicity (as shown in Figure 1.)

5.3 Upper bounding

In Steps 5 and 10, FIND-BUNDLING calls a function to upper bound the revenue obtainable in the subtree rooted at the node. In this paper we propose, and investigate the performance of, several such techniques. These techniques are discussed in each of the following subsubsections, respectively. If the technique indeed gives an upper bound (as opposed to sometimes giving a value that is below the actual revenue obtainable in the subtree rooted at the node), that is, the upper-bounding heuristic is *admissible*, FIND-BUNDLING will always give the optimal solution. We will also study their *monotonicity*, that is, whether the upper bound is non-increasing down each search path.[3]

WELFARE The first, and simplest, upper-bounding technique is to use the highest achievable welfare, constrained to honoring the bundling from the path so far. Specifically, the technique generates a set of "items" M' consisting of the *bundles* created so far in the search, and the remaining items unbundled, and then calls DETERMINE-WINNERS on the "item" set. We call this heuristic WELFARE.

PROPOSITION 5. WELFARE *is admissible and monotonic.*

PROOF. We prove monotonicity first. In the computation of WELFARE, all yet-undecided items at the search node are unbundled. So, WELFARE corresponds to the welfare of the finest bundling achievable in that subtree. Naturally, the optimal welfare is nondecreasing as we make strictly finer bundlings. It follows that WELFARE is monotonic.

We prove admissibility next. The welfare at a node upper bounds the revenue at the node. For any descendant d of the current node, we have that WELFARE at d upper bounds the revenue at d. From monotonicity we have that WELFARE at the current node is no less than WELFARE at d. Thus WELFARE at the current node is an upper bound. □

VCG$^+$ Our second upper-bounding technique is like computing VCG payments for the bundling at the node but with the negative term chosen so as to maximize payments (under the condition that no bidder pays more than her valuation). We call this VCG$^+$.

Let us now formalize this idea. The sum of payments like the VCG payments for the bundling at the node, but with the negative term chosen so as to maximize the payments, is

$$\max_{\alpha \in A_\phi} \sum_{i \in N} \sum_{j \neq i} [v_j(\alpha^*_{-i}) - v_j(\alpha)]$$

Here, $\alpha^*_{-i} \in A_\phi$ is the optimal allocation without bidder i, for the bundling, ϕ, at the node. A_ϕ is the set of allocations consistent with bundling ϕ. Also, $\alpha \in A_\phi$ is any allocation that satisfies the bundling at the node. We further tighten this upper bound by making sure that no bidder is charged more than her valuation. This gives us the formula for our upper-bounding technique VCG$^+$:

$$\max_{\alpha \in A_\phi} \sum_{i \in N} \min \left\{ \sum_{j \neq i} [v_j(\alpha^*_{-i}) - v_j(\alpha)],\ v_i(\alpha) \right\} \quad (19)$$

In the special case where α is the welfare-maximizing allocation at the current node, this equals the revenue of running VCG at the node.

The MIP in Figure 3 implements this idea. Constraints 21 and 22 ensure that each bidder pays her VCG$^+$ payment. Constraints 23 and 24 ensure that only one bidder can win each bundle in ϕ_p, the bundling at the node. Constraints 25 and 26 ensure that each bidder wins only one of his bids.

$$\max \sum_{i=1}^{n} p_i \quad (20)$$

$$p_i \leq \sum_{j \in B_i} v_j \cdot I_j \qquad \forall i \in N \quad (21)$$

$$p_i \leq \sum_{j \in B_{-i}} v_j \cdot I_j^{-i} - \sum_{j \in B_{-i}} v_j \cdot I_j \quad \forall i \in N \quad (22)$$

$$\sum_{j \in B_{\sqcap b}} I_j \leq 1 \qquad \forall b \in \phi_p \quad (23)$$

$$\sum_{j \in B_{\sqcap b}} I_j^{-i} \leq 1 \qquad \forall i \in N, b \in \phi_p \quad (24)$$

$$\sum_{j \in B_i} I_j \leq 1 \qquad \forall i \in N \quad (25)$$

$$\sum_{j \in B_k} I_j^{-i} \leq 1 \qquad \forall i, k \in N \quad (26)$$

Figure 3: Mixed integer program for VCG$^+$. Here, $B_{\sqcap b} = \{j \in B | S_j \sqcap b \neq \emptyset\}$ is the set of all bids that are interested in some bundle b.

We now prove that VCG$^+$ gives an upper bound to the revenue found at any node in the subtree rooted at the current node.

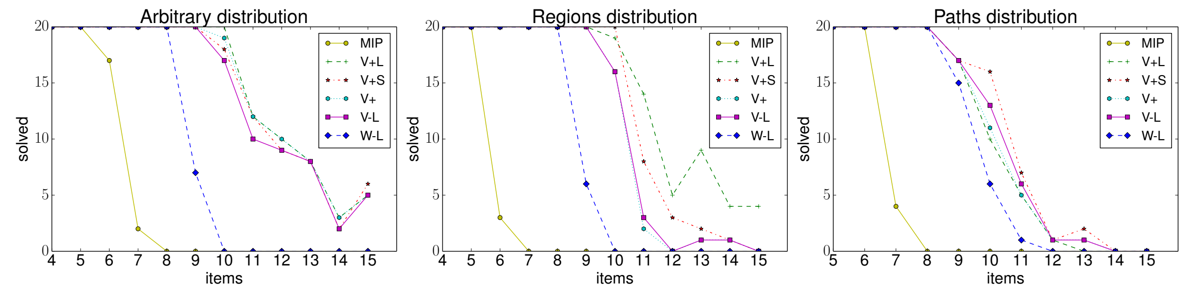
PROPOSITION 6. VCG$^+$ *is admissible and monotonic.*

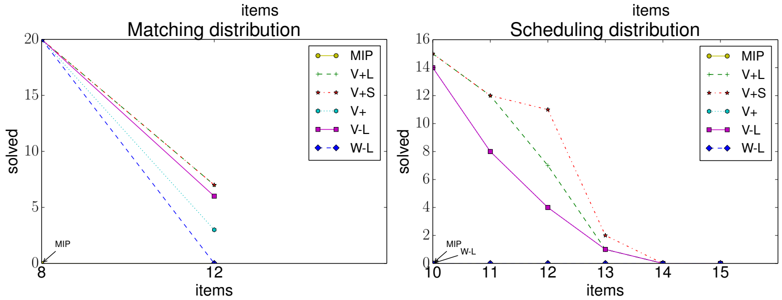
PROOF. Consider an arbitrary current node p. We prove admissibility first. VCG$^+$ selects the best set of winning bids from all legal allocations of winning bids for the bundling ϕ_p. For any descendant d, we get a bundling ϕ_d such that if two items $a, b \in M$ are bundled together in ϕ_p then they are also bundled together in ϕ_d. This means that any valid set of winning bids at d is also a valid set of winning bids at p. In particular, the welfare-maximizing allocation at d is a valid set of winning bids at p; call this allocation α_d. Since α_d is a valid allocation for VCG$^+$at p, we just need to show that the payment that VCG$^+$can obtain at p by selecting this allocation is no smaller than $r_d = \sum_{i=1}^{n}[W^{*,-i}_{\phi_d,B} - W^{*,-i}_{\phi_{d,-i},B}]$, the VCG revenue of ϕ_d. This is true because the negative term is the same (since the allocations are the same), and for the positive term $W^{*,-i}_{\phi_d,B}$, we have $W^{*,-i}_{\phi_d} \leq W^{*,-i}_{\phi_p}$ since optimal welfare is nonincreasing with more bundling.

Monotonicity follows from the fact that the VCG$^+$ MIP for any descendant of d of p is the VCG$^+$ MIP for p with additional constraints added. Adding constraints cannot increase the value. □

VCG$^+_{LB}$ We introduce a technique that tightens the bound of VCG$^+$ based on the observation that any constraint that does not cut off any of the welfare-maximizing allocations at any node in the subtree will preserve the upper bound. With this in mind, we add the constraint $\sum_{j \in B} v_j \cdot I_j \geq W_{LB}$ to the MIP of Figure 3, where W_{LB} is a lower bound on the welfare found at any node in the subtree.

[3]Unlike in typical tree (or graph) search in artificial intelligence, here monotonicity does not imply admissibility because there is no notion of the cost of the path from the root to the node.





Distribution	MIP	V+L	V+S	V+	V-L	W-L
Arbitrary	59	178	175	177	171	107
Regions	43	175	154	140	141	106
Paths	44	113	123	115	118	102
Matching	0	27	27	23	26	20
Scheduling	0	35	40	27	27	0
Total	146	528	519	482	483	335

Figure 4: Plots showing the number of instances solved for varying numbers of items with the MIP, and various heuristic configurations for our branch-and-bound approach. The table shows the total number of instances solved for each algorithm and distribution pair.

At each search node, we use two different ways of computing a value for W_{LB}, and we use the larger of the two. The first is the maximum valuation of any single bid, which is obviously a lower bound since we can always let any single bid be the only winner. The second is obtained by running DETERMINE-WINNERS on the bids that do not use any of the yet-undecided items. This is clearly a lower bound on the welfare of any node in that subtree because the allocation that it finds remains feasible for all nodes the subtree.

PROPOSITION 7. VCG^+_{LB} *is admissible and monotonic.*

PROOF. Admissibility follows immediately from the admissibility of VCG^+ and the fact that both ways of computing W_{LB} indeed yield lower bounds as argued in Section 5.3.

Monotonicity follows from monotonicity of VCG^+ and the fact that both ways of computing W_{LB} yield W_{LB} values that are non-decreasing down each search path (because the set of bids to choose from grows or stays the same). □

5.4 Variable ordering

In Step 6, the function BRANCHINGITEM(node) chooses the item to branch on at the node. Any choice will yield a correct algorithm, but some choices lead to smaller trees than others and thus shorter run times. The motivation for our variable-selection heuristic is that we want to pick an item that most likely needs to be bundled so that we get a fairly balanced search tree (where the promising branches are the many branches where this item is bundled). In contrast, branching on an item that likely should not be bundled would render the one "unbundling" branch the most promising and would thus yield lopsided deep trees.

We introduce two branching heuristics, PRICE-GAP-SIZE and PRICE-GAP-LOG. They both work by first computing the highest and second-highest "normalized bid price" for each item. Then, the item that has the greatest difference between the highest and second-highest normalized bid price is chosen for branching. The idea is that such items are promising for bundling because there is not enough competition on them. PRICE-GAP-SIZE In PRICE-GAP-SIZE, we use the following formula for normalized bid price: $\frac{v_j}{|S_j|}$. In PRICE-GAP-LOG, we use the following formula for normalized bid price: $\frac{v_j}{log(|S_j|)}$.

Using the number of items in the bid, $|S_j|$, for normalization gives a more precise estimate of the valuation of each item, but using the logarithm of $|S_j|$ favors bids with a greater number of items, which can lead to more important decisions being made early. Logarithmic normalization has been experimentally shown to perform well in the winner determination problem [39] so we included that in our experiments as well.

6. EXPERIMENTS

We conducted experiments with all our different algorithmic approaches using the Combinatorial Auction Test Suite (CATS) [26], which is the leading combinatorial auction benchmark suite. We generated a test suite from all the CATS distributions (excluding the old, unrealistic "legacy" distributions): arbitrary, matching, paths, regions, and scheduling. For each distribution we generated instances with 4-15 items, and bids equal to 0.2, 0.5, 1, 2, 5, and 10 times the number of items, with 20 instances generated for each of these settings. For space reasons we include only the most interesting results here. All experiments were conducted on a cluster, with each experiment run on a core on AMD Opteron quad-core 2.0GHz processors and 10GB of RAM available. The operating system was Rocks Version 6.1. The MIP models were solved using CPLEX 12.5. We used a time limit of 15 minutes for each run.

In this paper we present the most interesting results from our experiments. For the interested reader, a full description and results for all our experiments can be found in a companion tech report that will be made available upon publication.

In addition to the techniques described above, we evaluated a node-ordering technique we call LAZY-BOUND, where the nodes' upper bounds are computed as the nodes are taken from the open list. This had a negligible positive impact on performance, and is discussed in the companion tech report. All algorithms presented here use the technique. We also designed and evaluated an upper-bounding technique we call EXTERNALITY-FLOW. Its performance was worse than that of VCG^+ and VCG^+_{LB}, its discussion is also relegated to the companion tech report.

Figure 4 shows five plots giving the number of instances solved by our various algorithmic configurations. A plot is given for each of the five CATS distribtutions. For each plot, the performance of the MIP and five heuristic configurations for FIND-BUNDLING are shown. The FIND-BUNDLING configurations are as follows: VCG^+_{LB} and PRICE-GAP-LOG(V+L), VCG^+_{LB} and PRICE-GAP-SIZE (V+S), VCG^+_{LB} with no variable ordering heuristic (V+), VCG^+ and PRICE-GAP-LOG (V-L), and WELFARE and PRICE-GAP-LOG (W-L). Finally, the table in Figure 4 shows the total number of instances solved for each distribution and algorithm pair.

The presented results are for using instances where we generate approximately ten times as many bids as there are items. The results are representative of algorithm performance for smaller numbers of bids, but these were hardest and thus best showed the performance difference. In the following sections we discuss the results in light of the various heuristics we developed.

MIP vs. our custom algorithm For the basic MIP approach, our experiments show that it is unable to scale beyond 7 items for all of the distributions (Figure 4). In contrast to this, our best algorithms (V+L and V+S) solve instances with up to 15 items. We also see that the total number of instances solved (table in Figure 4) by V+L and V+S is far greater than for the MIP approach.

Upper bounding heuristics We conducted experiments with each of the three upper-bounding techniques (WELFARE, VCG^+, and VCG^+_{LB}). All three use PRICE-GAP-LOG for variable ordering.

On the arbitrary and regions distributions, we see that VCG^+_{LB} is able to solve at least 3 out of 20 instances for all item sizes, and more than 10 for item sizes 11 and lower. On these distributions, VCG^+ performs almost as well, with only 7 fewer instances solved overall on the arbitrary distribution and 34 fewer on the regions distribution. WELFARE performs significantly worse on these two distributions, solving approximately 70 instances less on each. For the matching distribution, all algorithms have somewhat similar performance, but the relative order of the upper-bounding heuristics is the same. VCG^+_{LB} solves 27 out of 40 instances, whereas the MIP solves none. For the paths and scheduling distributions, all the upper-bounding techniques perform significantly worse. VCG^+_{LB} and VCG^+ only scale to 13 items, but still significantly outperform WELFARE, which scales to 11 items. Interestingly, the total number of instances solved is slightly higher for V-L than for V+L. Overall, VCG^+_{LB} and VCG^+ clearly outperform WELFARE by a significant margin. VCG^+_{LB} also outperforms VCG^+ on 4 out of 5 distributions, 2 by a significant margin.

Variable-ordering heuristics We experimented with all three settings for the variable ordering heuristic while using VCG^+_{LB} for upper bounding: PRICE-GAP-LOG (V+L) , PRICE-GAP-SIZE (V+S), and no heuristic (V+). The performance on the arbitrary and matching distributions is almost the same for all three heuristics, with PRICE-GAP-LOG solving three more instances than PRICE-GAP-SIZE. For the regions distribution, we see a more significant performance difference. PRICE-GAP-LOG solves 175 instances, where PRICE-GAP-SIZE solves 154 instances, and using no heuristic solves 140. Finally, for the paths and scheduling distributions, PRICE-GAP-SIZE performs somewhat better than PRICE-GAP-LOG. On paths, it solves 123 instances while PRICE-GAP-LOG solves 113. On scheduling, it solves 40 while PRICE-GAP-LOG solves 35. Overall, PRICE-GAP-LOG solved 9 instances more than PRICE-GAP-SIZE, both clearly outperforming not using a variable-ordering heuristic. However, even PRICE-GAP-LOG is outperformed by not using a variable-ordering heuristic on one distribution (although marginally).

6.1 Revenue increase and surplus extraction

Finally, we conducted experiments that empirically examine how well our approach both bridges the gap to the optimal revenue, and how much it improves over the VCG revenue. For low bids-to-items multipliers, the revenue increase was very often infinitely higher. This is because VCG often received no revenue in these settings, whereas VCG with optimal bundling usually achieves at leaft half the social welfare in revenue. These results are not shown here as they are trivially stated.

#items	arbitrary	regions	paths	matching	scheduling
4	5.9 / 89	4.6 / 91	- / -	- / -	- / -
5	1.6 / 90	7.6 / 87	0.6 / 89	- / -	- / -
6	4.4 / 90	0 / 82	1.7 / 86	- / -	- / -
7	6.3 / 87	0.1 / 84	7.6 / 85	- / -	- / -
8	9 / 88	1.6 / 84	2.2 / 87	3.4 / 84	- / -
9	2.6 / 92	4.7 / 91	3.9 / 88	- / -	- / -
10	2.2 / 91	4.1 / 92	1.2 / 89	- / -	5.4 / 88
11	3.8 / 92	- / -	3 / 91 11	- / -	5.1 / 88
12	8.6 / 92	- / -	- / -	8.4 / 89	3 / 86

Figure 5: The first column for each distribution shows the percentage increase in revenue from bundling compared to regular VCG with no bundling. The second shows the percentage of social welflare extracted as revenue.

Table 5 shows the results from running optimal bundling in the VCG, VCG (with no bundling), and welfare computation on each of the five CATS distributions, when the number of bids is 5 times the number of items. The percentages given are average increases over 10-20 instances for each parameter setting. Entries with - / - denote that not enough instances were generated or solved.

For the bids-to-items ratio 5 (Figure 5), the revenue increase over VCG tends to lie in the 2-20% range. About 90% of the social welfare is obtained in revenue for these settings.

In real-life applications, the revenue increase is likely to be even higher, as CATS tends to generate bids that span all items, with valuation for such bids being close to the optimal welfare, and thus causing artificially high compeition across all items. Similarly, CATS only generates undominated bids, that is, bids whose value is higher than what could be obtained in welfare from the items in the bid using other bids that only want those items. This means that CATS generates artificially strong competition, and thus revenue, for the given number of bids and items.

7. CONCLUSION AND FUTURE WORK

We studied bundling, a common revenue-enhancement approach, in the context of the most commonly studied combinatorial auction mechanism, VCG, adopting a point prior model. We proved robustness to inaccuracy in the prior, and showed that computing the optimal bundling is NP-hard even with a point prior. Then, we presented a custom branch-and-bound framework for finding the optimal bundling. In that framework, we introduced several techniques for branching, upper bounding, lower bounding, and lazy bounding. Experiments on CATS distributions validated the approach and showed that our techniques dramatically improve scalability over a leading MIP solver.

There are many interesting directions for future research. Affine maximizer auctions, virtual valuations combinatorial auctions, and λ-auctions all support unlimited, bidder-specific reserve prices. In contrast, we studied bundling alone as a revenue-enhancement tool. There are many interesting questions about the relative power of different forms of bundling, different forms of reserve pricing, and combinations thereof— both in the auction and catalog sales contexts. Second, we plan to extend our algorithms to settings with structure. For example, in many advertising markets, the inventory segments are defined by vectors of attributes, and that can, in some settings, provide additional structure.

Acknowledgements. This material is based on work supported by the National Science Foundation under grants CCF-1101668, IIS-0964579, and IIS-1320620.

References

[1] William James Adams and Janet L. Yellen. "Commodity Bundling and the Burden of Monopoly". In: *Quarterly Journal of Economics* 90.3 (1976), pp. 475–498.

[2] Mark Armstrong. "Multiproduct Nonlinear Pricing". In: *Econometrica* 64.1 (1996), pp. 51–75.

[3] Mark Armstrong. "Optimal Multi-Object Auctions". In: *Review of Economic Studies* 67 (2000), pp. 455–481.

[4] Lawrence M. Ausubel and Paul Milgrom. "The Lovely but Lonely Vickrey Auction". In: *Combinatorial Auctions*. Ed. by Peter Cramton, Yoav Shoham, and Richard Steinberg. MIT Press, 2006. Chap. 1.

[5] Christopher Avery and Terrence Hendershott. "Bundling and Optimal Auctions of Multiple Products". In: *Review of Economic Studies* 67 (2000), pp. 483–497.

[6] Yannis Bakos and Erik Brynjolfsson. "Bundling Information Goods: Pricing, Profits, and Efficiency". In: *Management Science* 45.12 (1999), pp. 1613–1630.

[7] Michael Benisch and Tuomas Sandholm. "A Framework for Automated Bundling and Pricing Using Purchase Data". In: *Proceedings of the Conference on Auctions, Market Mechanisms and Their Applications (AMMA)*. 2011.

[8] D Berend and T Tassa. "Improved Bounds on Bell Numbers and on Moments of Sums of Random Variables". In: *Probability and Mathematical Statistics* 30.2 (2010), pp. 185–205.

[9] Chirstopher H. Brooks and Edmund H. Durfee. "Toward Automated Pricing and Bundling of Information Goods". In: *AAAI Workshop on Knowledge-based Electronic Markets*. 2000, pp. 8–13.

[10] Yang Cai, Constantinos Daskalakis, and Matt Weinberg. "Optimal Multi-Dimensional Mechanism Design: Reducing Revenue to Welfare Maximization". In: *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*. 2012.

[11] I Chakraborty. "Bundling decisions for selling multiple objects". In: *Economic Theory* 13 (1999), pp. 723–733.

[12] Ed H. Clarke. "Multipart pricing of public goods". In: *Public Choice* 11 (1971), pp. 17–33.

[13] Vincent Conitzer and Tuomas Sandholm. "Complexity of Mechanism Design". In: *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. Edmonton, Canada, 2002, pp. 103–110.

[14] Vincent Conitzer and Tuomas Sandholm. "Applications of Automated Mechanism Design". In: *UAI-03 workshop on Bayesian Modeling Applications*. Acapulco, Mexico, 2003.

[15] Vincent Conitzer and Tuomas Sandholm. "Self-interested Automated Mechanism Design and Implications for Optimal Combinatorial Auctions". In: *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*. New York, NY, 2004, pp. 132–141.

[16] Robert E. Dansby and Cecilia Conrad. "Commodity Bundling". In: *The American Economic Review* 74.2 (1984), pp. 377–381.

[17] Robert Day and Paul Milgrom. "Core-selecting package auctions". In: *International Journal of Game Theory* 36.3 (2008), pp. 393–407.

[18] Arpita Ghosh, Hamid Nazerzadeh, and Mukund Sundararajan. "Computing optimal bundles for sponsored search". In: *International Workshop On Internet And Network Economics (WINE)*. 2007, pp. 576–583.

[19] Theodore Groves. "Incentives in Teams". In: *Econometrica* 41 (1973), pp. 617–631.

[20] Joseph P. Guiltinan. "The Price Bundling of Services: A Normative Framework". In: *Journal of Marketing* 51.2 (1987), pp. 74–85.

[21] Ward Hanson and R. Kipp Martin. "Optimal bundle pricing". In: *Manage. Sci.* 36.2 (1990), pp. 155–174.

[22] Lorin M. Hitt and Pei-yu Chen. "Bundling with Customer Self-Selection: A Simple Approach to Bundling Low-Marginal-Cost Goods". In: *Management Science* 51.10 (2005), pp. 1481–1493.

[23] Ali Hojjat, John Turner, Suleyman Cetintas, and Jian Yang. "Delivering Guaranteed Display Ads under Reach and Frequency Requirements". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2014.

[24] Philippe Jehiel, Moritz Meyer-Ter-Vehn, and Benny Moldovanu. "Mixed bundling auctions". In: *Journal of Economic Theory* 127.1 (2007), pp. 494–512.

[25] Jeffrey O. Kephart, Christopher H. Brooks, Rajarshi Das, Jeffrey K. MacKie-Mason, Robert Gazzale, and Edmund H. Durfee. "Pricing information bundles in a dynamic environment". In: *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*. Tampa, Florida, USA: ACM Press, 2001, pp. 180–190.

[26] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. "Towards a Universal Test Suite for Combinatorial Auction Algorithms". In: *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*. Minneapolis, MN, 2000, pp. 66–76.

[27] Anton Likhodedov and Tuomas Sandholm. "Methods for Boosting Revenue in Combinatorial Auctions". In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. San Jose, CA, 2004, pp. 232–237.

[28] Anton Likhodedov and Tuomas Sandholm. "Approximating Revenue-Maximizing Combinatorial Auctions". In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. Pittsburgh, PA, 2005.

[29] Alejandro M Manelli and Daniel R Vincent. "Bundling as an optimal selling mechanism for a multiple-good monopolist". In: *Journal of Economic Theory* 127.1 (2006), pp. 1–35.

[30] Eric Maskin and John Riley. "Optimal Multi-Unit Auctions". In: *The Economics of Missing Markets, Information, and Games*. Ed. by Frank Hahn. Clarendon Press, Oxford, 1989. Chap. 14, pp. 312–335.

[31] R. Preston McAfee, John McMillan, and Michael D. Whinston. "Multiproduct Monopoly, Commodity Bundling, and Correlation of Values". In: *Quarterly Journal of Economics* 104.2 (1989), pp. 371–383.

[32] Roger Myerson. "Optimal auction design". In: *Mathematics of Operation Research* 6 (1981), pp. 58–73.

[33] Thomas Palfrey. "Bundling Decisions by a Multiproduct Monopolist with Incomplete Information". In: *Econometrica* 51 (1983), pp. 463–484.

[34] Baharak Rastegari, Anne Condon, and Kevin Leyton-Brown. "Revenue monotonicity in deterministic, dominant-strategy combinatorial auctions". In: *Artif. Intell.* 175.2 (2011), pp. 441–456.

[35] Paat Rusmevichientong, Benjamin Van Roy, and Peter W. Glynn. "A Nonparametric Approach to Multiproduct Pricing". In: *Oper. Res.* 54.1 (2006), pp. 82–98. ISSN: 0030-364X.

[36] Tuomas Sandholm. "Algorithm for Optimal Winner Determination in Combinatorial Auctions". In: *Artificial Intelligence* 135 (Jan. 2002), pp. 1–54.

[37] Tuomas Sandholm. "Automated Mechanism Design: A New Application Area for Search Algorithms". In: *International Conference on Principles and Practice of Constraint Programming (CP)*. Cork, Ireland, 2003, pp. 19–36.

[38] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. "Coalition Structure Generation with Worst Case Guarantees". In: *Artificial Intelligence* 111.1–2 (1999), pp. 209–238.

[39] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. "CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions". In: *Management Science* 51.3 (2005), pp. 374–390.

[40] Richard Schmalensee. "Gaussian Demand and Commodity Bundling". In: *The Journal of Business* 57.1 (1984), S211–S230.

[41] George J. Stigler. "United States v. Loew's Inc.: A Note on Block-Booking". In: *The Supreme Court Review* 1963 (1963), pp. 152–157.

[42] John Turner. "The Planning of Guaranteed Targeted Display Advertising". In: *Operations Research* 60 (1 2012), pp. 18–33.

[43] William Vickrey. "Counterspeculation, Auctions, and Competitive Sealed Tenders". In: *Journal of Finance* 16 (1961), pp. 8–37.

[44] William Walsh, Craig Boutilier, Tuomas Sandholm, Robert Shields, George Nemhauser, and David Parkes. "Automated Channel Abstraction for Advertising Auctions". In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. 2010.

[45] Shin-yi Wu, Lorin M. Hitt, Pei-yu Chen, and G. Anandalingam. "Customized Bundle Pricing for Information Goods: A Nonlinear Mixed-Integer Programming Approach". In: *Management Science* 54.3 (2008), pp. 608–622.