2 Combinatorial Auctions and the VCG Mechanism

2.1 Combinatorial Auctions

Recall that the VA is concerned with auctioning off a single good. Combinatorial auctions are motivated by the following natural question.

Question 2.1 What if there is a set S of $m > 1$ goods to be auctioned off to n players? How can we extend the VA to this more general setting?

A natural idea is to run a separate Vickrey auction for each of the m goods. This works (i.e., properties (P1)–(P4) hold) if each player i has a separate value for each item, and the value of a subset $T \subseteq S$ of goods to player i is the sum of its values for the goods of T. (Exercise: check this.) Unfortunately, this simple approach ignores the possible dependencies between the outcomes of the different auctions for players. More specifically, it ignores:

(1) *substitutes:* a player's value of getting (say) two goods is less than the sum of its values for each individually (e.g., they are at least partially redundant);

(2) *complements:* a player's value of getting (say) two goods is greater than the sum of its values for each individually (e.g., they are at least partially co-dependent).

Indeed, one of the applications that kicked off the systematic study of combinatorial auctions was the problem (faced by the FAA) of auctioning off take-off and landing slots at airports to the major airlines. Two take-off slots from the same airport at almost the same time are substitutes from an airline's perspective, whereas a take-off slot at one airport and a landing slot at a second airport (at the appropriate subsequent time) act as complements.

Informally, a *combinatorial auction (CA)* is an auction that allocates a set of many goods to bidders in the presence of substitutes and complements. As we will see, designing good combinatorial auctions is much more challenging than designing good auctions for selling a single good.

2.2 The VCG Mechanism

Our first combinatorial auction is a classical, powerful mechanism called the VCG mechanism. (By "mechanism", we essentially mean some sort of incentive-compatible protocol.) The "V" stands for Vickrey [11], the "C" for Clarke [1], and the "G" for Groves [2], three researchers who gave successively more general versions of Vickrey auction. The good news about the VCG mechanism is that it satisfies all of properties (P1)–(P3) from Subsection 1.3 (incentive constraints, economic efficiency, and general valuations). The bad news is that it is highly computationally intractable.

To specify the VCG mechanism, we first need to say what we mean by a "valuation" of a player i when there is a set S of $m > 1$ goods. For now, we will allow a very general definition; later we will look at several special cases. We call a subset $T \subseteq S$ of goods a *bundle*. The *valuation* v_i of the player i is a function from the set 2^S of all possible bundles to the nonnegative reals. In other words, the valuation specifies the value $v_i(T)$ of player i of every conceivable bundle $T \subseteq S$ of goods that it might receive. Note that with m goods, there are 2^m such bundles. We assume that $v_i(\emptyset) = 0$ for every i, though this is not an essential assumption. For this section, we do not even need to assume that v_i is nonnegative or that it is monotone (i.e., that $T \subseteq T'$ implies $v_i(T) \le v_i(T')$), though we will make these assumptions in future sections.

Note that such valuations are certainly expressive enough to model substitutes and complements. For example, if $S = \{1, 2\}$ contains two goods which are perfect substitutes for a player i, then i's valuation might be $v(\{1\}) = v(\{2\}) = v(\{1, 2\}) = 1$. If the two goods are complements, then i's valuation might be given by $v(\{1\}) = v(\{2\}) = 0$ and $v(\{1, 2\}) = 1$.

Recall that for a single-item auction, the job of the auction is to determine a winner and what price to charge. In a combinatorial auction there can be multiple "winners"—the outcome of a CA is to allocate a bundle $T_i \subseteq S$ to each player i such that bundles given to distinct players are disjoint (no good can be allocated to more than one winner). Accordingly, a CA can charge a different price p_i to each player i. As in the VA, we again assume quasilinear utilities, meaning:

- if player i receives the bundle T_i and is charged the price p_i, then its *utility* is $v_i(T_i) - p_i$.

We now state the VCG mechanism, deferring the description of the prices until Section 2.3. (Compare to the three steps of the VA.)

(1) Each player i submits a bid $b_i(T)$ for every possible non-empty bundle $T \subseteq S$. (We always implicitly assume that $b_i(\emptyset) = 0$.) (If the player is truthful, then $b_i(T) = v_i(T)$ for every $T \subseteq S$.)

(2) Choose an allocation $(T_1^*, \ldots, T_n^*)$ that maximizes

$$\sum_{i=1}^{n} b_i(T_i)$$

over all feasible allocations $\{T_i\}_{i=1}^n$ (feasible means that $T_i \cap T_j = \emptyset$ whenever $i \neq j$).

(3) Charge each player i an appropriate price p_i (to be determined).

Both steps (1) and (2) should alarm theoretical computer scientists—more on this shortly. Nevertheless, we can verify the properties (P2) and (P3) from Subsection 1.3 without even stating the prices.

Proposition 2.2 *The VCG mechanism is economically efficient. In other words, if all players bid truthfully, then the VCG mechanism outputs an allocation that maximizes*

$$\sum_{i=1}^{n} v_i(T_i)$$

over all feasible allocations.

Proof: Immediate from step (2) of the mechanism. ■

Proposition 2.3 *The VCG mechanism works with general valuations.*

Proof: By definition. ■

To discuss property (P4), we need to specify our criteria for computational tractability. Recall we are interested in auctions that run in polynomial time—but polynomial in what?

Question 2.4 Recall that merely specifying the valuation of a player requires $2^m - 1$ parameters. Should we be happy if an auction runs in time polynomial in this "input size"?

In this course, we will be ambitious: *our criteria for polynomial time will be polynomial in the number n of players and in the number m of goods.* In other words, we are only interested in CAs that scale reasonably with number of players and goods. The VCG mechanism clearly does not satisfy this stringent definition of computational tractability: merely communicating the bid of a single player in step (1) requires exponential resources. The VCG mechanism is also computationally inefficient in a second sense, as we will see in Section 3: even in special cases where bidders can communicate their entire valuation in polynomial time, the optimization problem that the VCG mechanism must solve in Step (2) can be highly intractable.

2.3 VCG Prices and Strategyproofness

To determine whether or not the VCG mechanism has property (P1) (i.e., satisfies incentive constraints), we must specify the prices charged in Step (3).

Question 2.5 Suppose we always set $p_i = 0$ for all i. Would this make the VCG mechanism truthful?

Question 2.6 We will give prices that generalize those in the VA. Can you think of what this would look like for a CA (say with two goods)?

We specify the VCG prices in a form due to Clarke [1]. In English, the definition is:

(A) set p_i equal to the damage caused to the other players by i's presence.

Mathematically, we have

$$p_i = \left(\max_{\{T_j\}_{j \neq i}} \sum_{j \neq i} b_j(T_j) \right) - \sum_{j \neq i} b_j(T_j^*), \tag{2}$$

where the maximum ranges over all feasible allocations of the goods to the $n-1$ players other than i (as usual, we insist that $T_j \cap T_k = \emptyset$ for all $j \neq k$). Several comments. First, to interpret these prices, it is often helpful to think of each of the bids $b_i(\cdot)$ in (2) as the corresponding true valuation $v_i(\cdot)$; after all, at the end of the day we will prove that the

VCG mechanism is truthful and thus expect bidders to bid their true valuations. (Of course, the price p_i cannot explicitly refer to a true valuation $v_i(\cdot)$ since these are unknown to the mechanism; it can only use the received bids as proxies for the true valuations.)

The first term on the right-hand side of (2) is the maximum-possible surplus if we delete player i's bid and optimize only for the $n-1$ other players. Note this is precisely the result of rerunning step (2) of the VCG mechanism after deleting i's bid from the input. Since player i did submit a bid, however, the VCG mechanism instead chose the allocation $\{T_j^*\}_{j=1}^n$ maximizing the surplus $\sum_{j=1}^n b_j(T_j^*)$ of all of the players. From the perspective of the $n-1$ players other than i, their collective benefit in this allocation is $\sum_{j \neq i} b_j(T_j^*)$, the second term on the right-hand side of (2). The right-hand side of (2) is therefore the extent to which the collective benefit of the $n-1$ players other than i would increase if player i was deleted and the VCG mechanism chose an allocation solely for their benefit—the damage caused to these players by i's presence.

The idea of these prices is to force a player to care about the welfare of the other players, thus aligning the objective of the player with the global objective of maximizing social surplus. This idea is common in economics and is often called "internalizing an externality".

Example 2.7 In the special case of a single-good auction, the price (2) specializes to the prices in the VA (0 for losers, the second-highest bid for the winner). To see this, note that with a single item, every bundle T_i^* has the form either $\emptyset$ (for losers) or $\{1\}$ (for the winner, where "1" denotes the item being sold). When a player submits a bid b_i in the VA, it corresponds to a bid $b_i(\{1\})$ in the current notation; as usual, we implicitly assume that $b_i(\emptyset) = 0$ for every player. Note also that step (2) of the VCG mechanism simply means giving the item to the highest bidder (which is step (2) of the VA).

First consider a player i that loses (so $T_i^* = \emptyset$ and $b_i(T_i^*) = 0$). Let k be the winner (so $T_k^* = \{1\}$ and $b_k(T_k^*)$ is its sealed bid b_k). The second term on the right-hand side of (2) is b_k. Since player i lost (i.e., did not have the highest bid), deleting the player and rerunning step (2) of the VA would still result in player k winning the item; the first term on the right-hand side of (2) is also b_k, resulting in a price $p_i = 0$ for player i. On the other hand, suppose player i wins the item, so $b_i(T_i^*) = b_i$ and $b_j(T_j^*) = 0$ for every $j \neq i$. The second term on the right-hand side of (2) is 0. If player i is deleted and step (2) of the VA is rerun, then the remaining player with the highest bid—the player that originally possessed the second-highest bid—wins, so the first term on the right-hand side of (2) is the second-highest bid, as in the VA.

The definition (2) of the VCG prices immediately gives the following.

Proposition 2.8 *VCG prices are nonnegative.*

Proof: One feasible solution for the maximization problem in the first term in the right-side of (2) is $\{T_j^*\}_{j \neq i}$ with total value $\sum_{j \neq i} b_j(T_j^*)$; the maximum can only be larger. ■

We now give a second definition and interpretation of the VCG prices. To obtain it, we simply add and subtract $b_i(T_i^*)$ from (2) and rearrange terms:

$$p_i = b_i(T_i^*) - \left[\sum_{j=1}^{n} b_j(T_j^*) - \left(\max_{\{T_j\}_{j \neq i}} \sum_{j \neq i} b_j(T_j) \right) \right]. \tag{3}$$

The way to think about (3) is that if the player i receives the bundle T_i^*, then it pays its bid $b_i(T_i^*)$ minus a discount (the expression in the square brackets in (3)). Note that the discount term is precisely the extent to which i's presence increases the maximum-achievable efficiency.

Question 2.9 What is the discount term in a single-item auction?

Recall that in a *first*-price single-item auction, there can be an incentive for players to underbid. (See the discussion following Proposition 1.3.) The rough intuition for the discount term above is that it simply gives players up front whatever they could gain by underbidding in a first-price version of the VCG mechanism.

From the second definition (3) of the VCG prices, we immediately obtain that the VCG mechanism is individually rational (recall Proposition 1.7).

Proposition 2.10 *The utility of a truthtelling bidder in the VCG mechanism is always nonnegative.*

Proof: The proposition is equivalent to showing that the discount term in (3) is always nonnegative. This holds because adding an extra bidder can only increase the maximum-achievable surplus (it only enlarges the set of feasible allocations). ■

All that remains to prove is that the VCG mechanism is truthful. (As an exercise, the reader is invited to prove that it is also strongly truthful in the sense of Proposition 1.6.)

Proposition 2.11 *The VCG mechanism is strategyproof. That is, for every player i, even if the player knows the full bids of all of the other players, player i maximizes its utility by bidding truthfully (setting $b_i(T) = v_i(T)$ for every non-empty bundle $T \subseteq S$).*

Proof: We follow the more general approach of Groves [2], which will make the proof of truthfulness more transparent. We first prove truthfulness for the wrong set of prices, and then show how to shift these prices to recover the VCG prices while maintaining truthfulness.

Modify the VCG mechanism so that in step (3) it computes the following price p_i for each player i:

$$p_i = -\sum_{j \neq i} b_j(T_j^*), \tag{4}$$

where as usual $\{T_j^*\}_{j=1}^n$ denotes the allocation computed in step (2) of the VCG mechanism. Note these are negative prices (i.e., subsidies) and are certainly not the VCG prices of (2) and (3). For example, for a single-item auction, these prices say that the winner should be charged nothing while all the losers should be paid the winner's bid! (Question: does this result in a strategyproof single-item auction?)

Note that the price (4) is defined so that any benefit to some other player also benefits the player i. More precisely, since player i's utility is its value for its bundle minus the price paid, its utility for a given allocation $\{T_j^*\}_{j=1}^n$ with the price in (4) is

$$v_i(T_i^*) + \sum_{j \neq i} b_j(T_j^*). \tag{5}$$

Suggestively, the VCG mechanism chooses in step (2) the allocation $\{T_j^*\}_{j=1}^n$ to maximize

$$\sum_{j=1}^n b_j(T_j) \tag{6}$$

over all feasible allocations $\{T_j\}_{j=1}^n$. Glibly, we might finish this part of the proof by saying that if player i bids truthfully, then the VCG mechanism's objective and its own are exactly aligned, which then results in an optimal outcome from i's perspective. While this argument is not incorrect, we proceed a bit more carefully.

As a sanity check, note that the only thing player i has control over is its bid $\{b_i(T)\}_{T \subseteq S}$. While the player cannot directly control the allocation $\{T_j^*\}_{j=1}^n$ chosen in step (2) of the VCG mechanism, it can potentially influence the choice of this allocation by varying its bid. Similarly, it cannot influence the functions $v_i(\cdot)$ and $b_j(\cdot)$ for $j \neq i$ (recall no collusion is allowed), only the allocation $\{T_j^*\}_{j=1}^n$ chosen by the VCG mechanism. Now, view (5) as an objective function for a discrete optimization problem (over allocations) from player i's perspective: there is some allocation, say $\{\widehat{T}_j\}_{j=1}^n$, that maximizes this function. The best-case scenario for player i is that some bid $\{b_i(T)\}_{T \subseteq S}$ coaxes the VCG mechanism into choosing this allocation $\{\widehat{T}_j\}_{j=1}^n$ as the allocation $\{T_j^*\}_{j=1}^n$ in its step (2)—if there exists such a bid, then no other bid can provide i with strictly more utility. But if player i bids truthfully ($b_i(T) = v_i(T)$ for all T), then the criterion (6) maximized by the VCG mechanism over all feasible allocations $\{T_j\}_{j=1}^n$ is

$$v_i(T_i) + \sum_{j \neq i}^n b_j(T_j), \tag{7}$$

and thus the VCG mechanism will indeed choose the allocation $\{\widehat{T}_j\}_{j=1}^n$ in step (2) (or some other allocation with equal value from player i's perspective). Thus player i maximizes its utility by bidding truthfully.

We have shown that the VCG mechanism is truthful provided we use the negative prices in (4). Here's the key idea of Groves [2]: suppose we shift each price p_i by a function $h_i(\{b_j\}_{j \neq i})$ that is *independent of i's bid b_i*. Here by independent we mean that once we fix all bids b_j for $j \neq i$, h_i is a constant function of b_i. In particular, it cannot depend on the allocation $\{T_j^*\}_{j=1}^n$ chosen by the VCG mechanism (which in turn is a function of b_i). For example, in the single-item case, $h_i(\{b_j\}_{j \neq i})$ could be the highest bid $\max_{j \neq i} b_j$ by some other player. Below, we use the standard shorthand b_{-i} to denote the set $\{b_j\}_{j \neq i}$ of bids by players other than i.

The claim is that adding such a function $h_i(\cdot)$ to the price p_i charged to player i does not affect strategyproofness. This follows from two simple facts. First, the new objective for player i (given fixed bids b_{-i} by the other players) is to choose a bid $\{b_i(T)\}_{T \subseteq S}$ to maximize

$$v_i(T_i^*) + \sum_{j \neq i} b_j(T_j^*) - c, \tag{8}$$

where c is the constant $h_i(b_{-i})$. Note that the sets of allocations maximizing (5) and (8) are exactly the same. Second, the allocation $\{T_j^*\}_{j=1}^n$ chosen by the VCG mechanism is independent of the prices (and of h_i in particular), and depends only on the bids. Thus bidding truthfully still causes the VCG mechanism to choose an allocation that maximizes (8) over all feasible allocations. This completes the proof of the claim.

Finally, note that instantiating

$$h_i(b_{-i}) = \max_{\{T_j\}_{j \neq i}} \sum_{j \neq i} b_j(T_j)$$

for each player i gives the VCG prices (2). ■

2.4 Summary

In this section we described the classical VCG mechanism for CAs. (The mechanism can also be defined much more generally; see [7].) On the plus side, it has properties (P1)–(P3) from Subsection 1.3: it satisfies both incentive constraints and economic efficiency even with general valuations. Unfortunately, it is computationally intractable—even the bidding step (step (1)) requires an exponential (in m) amount of communication (and time).

3 Single-Minded Bidders

The VCG mechanism has all of the properties that we'd want of a CA except for computational tractability. In this section we begin exploring the following question, which has been systematically studied only relatively recently (since the late 1990s, mostly by computer scientists): *how much do we need to relax the properties (P1)–(P3) of Subsection 1.3 to recover computational tractability (P4)?* We have already noted that if we weaken (P3) by assuming that bidders' valuations have no complements or substitutes, then we can easily achieve the other three properties by running a separate Vickrey auction for each good (see the discussion following Question 2.1). What can we accomplish with (at least some degree of) complements and/or substitutes?

3.1 Preliminaries

In this section we will focus on a highly restricted class of valuations, which essentially model an extreme form of complements.

Definition 3.1 Let S be a set of goods and i a bidder with valuation v_i. The bidder i is *single-minded* if there is a set $A_i \subseteq S$ of goods and a value $\alpha_i \geq 0$ such that:

(a) $v_i(T_i) = \alpha_i$ whenever $T_i \supseteq A_i$; and

(b) $v_i(T_i) = 0$ otherwise.

Thus from i's perspective there are only two distinct outcomes: either it gets all of the goods it wants (the set A_i), in which case its value for its bundle is α_i, or it fails to get all of these goods, in which case its value for its bundle is 0.

The motivation for this definition is twofold. First, it is a conceptually simple type of valuation that nevertheless models one of the quintessential aspects of CAs (complements). Second, it immediately gets rid of the initial computational stumbling block for the VCG mechanism: now players' valuations can be implicitly but completely specified in time polynomial in n and m, since each player i can simply report (proxies for) its set A_i and value α_i. We should therefore ask the following.

Question 3.2 For the special case of single-minded bidders, can the VCG mechanism be implemented to run in polynomial time?

If the answer is "yes", then we can move on to more general classes of valuations; if the answer is "no", then we will need to design a new (computationally tractable) mechanism even for the case of single-minded bidders.

The answer to Question 3.2 is no (assuming $P \neq NP$). The reason is that the VCG mechanism is computationally inefficient in two distinct senses. First, as we have repeatedly noted, the bidding step (1) requires exponential communication (for general valuations). Second, even when this problem is assumed away (as with single-minded bidders), the allocation step (2) of VCG can require exponential *computation*.

Precisely, consider the optimization problem of maximizing the surplus (1), given the true valuations of the bidders. This problem is typically called the *winner determination (WD)* problem. Note that step (2) of the VCG mechanism is precisely the WD problem (where bids are used as surrogates for true valuations). For single-minded bidders, the WD problem has the following form: given the valuations (truthful bids) of the players, as specified by the pairs $(A_1, \alpha_1), \ldots, (A_n, \alpha_n)$, grant a set of disjoint bids (i.e., a subset of players such that the corresponding A_i's are pairwise disjoint) to maximize the sum $\sum \alpha_i$ of the values of the granted bids. We next show that the WD problem is hard, even in the special case of single-minded bidders.

Proposition 3.3 ([5, 9]) *The WD problem for single-minded bidders is NP-hard.*

Proof: By a reduction from the NP-hard problem Weighted Independent Set (WIS). Given an instance of WIS, specified by a graph $G = (V, E)$ and a weight w_v for each vertex $v \in V$, construct the following instance of the WD problem: the set of goods is the set E of edges of G; the set of players is the set V of vertices; for a vertex/player $v \in V$, set $\alpha_v = w_v$ and A_v equal to the set of edges of G that are incident to v. A subset of vertices/players is then a

WIS of G if and only if it is a subset of bids that can be simultaneously granted. Moreover, this bijective correspondence preserves the total weight/value of the solution. ■

Unfortunately, WIS is not just an NP-hard problem; it is a "really hard" NP-hard problem. To make this precise, recall that a ρ-approximation algorithm for a maximization problem is a polynomial-time algorithm that always recovers at least a $1/\rho$ fraction of the value of an optimal solution. (By our convention, ρ is always at least 1.)

Fact 3.4 ([3]) *For every $\epsilon > 0$, there is no $O(n^{1-\epsilon})$-approximation algorithm for WIS, where n denotes the number of vertices (unless $NP \subseteq ZPP$).*

Fact 3.4 basically says that the WIS problem admits no non-trivial approximation algorithm. (Note that simply picking the max-weight vertex gives an n-approximation for WIS.) More relevant for CAs is the following consequence of Fact 3.4.

Corollary 3.5 *For every $\epsilon > 0$, there is no $O(m^{\frac{1}{2}-\epsilon})$-approximation algorithm for WIS, where m denotes the number of edges (unless $NP \subseteq ZPP$).*

Corollary 3.5 follows from Fact 3.4 because the number of edges of a (simple) graph is at most quadratic in the number of vertices.

Because the reduction in the proof of Proposition 3.3 is "approximation preserving" (it gives a bijection that preserves the objective function values of corresponding solutions of WIS and WD), it implies the following strong negative result about approximating the WD problem with single-minded bidders.

Corollary 3.6 *For every $\epsilon > 0$, there is no $O(m^{\frac{1}{2}-\epsilon})$-approximation algorithm for WD with single-minded bidders, where m denotes the number of goods (unless $NP \subseteq ZPP$).*

The upshot of Corollary 3.6 is rather bleak: if we want a polynomial-time CA—property (P4) from Subsection 1.3—then even if we assume single-minded bidders (sacrificing significant valuation generality (P3)), and even if we ignore incentive-compatibility (P1), then we must take a big hit on property (P2) and settle for (at best) an $O(\sqrt{m})$-approximation of the surplus.

At least the bad news stops here: we next design a CA for single-minded bidders that is poly-time implementable, achieves the best-possible approximation of the surplus under this constraint ($O(\sqrt{m})$), and also satisfies the incentive constraints (P1). We present this CA in two parts: first, we present a poly-time $O(\sqrt{m})$-approximation algorithm for WD with single-minded bidders (Subsection 3.2); then we show how to charge prices to turn this WD algorithm into an incentive-compatible mechanism (Subsection 3.3).

3.2 Approximate Winner Determination

We now design an approximation algorithm for the following problem: given a set S of m goods and (truthful) bids $(A_1, \alpha_1), \ldots, (A_n, \alpha_n)$, which bids should we grant to maximize the total value of granted bids? (Here by "grant bid (A_i, α_i)" we mean assign player i the bundle $T_i = A_i$; obviously granted bids should be pairwise disjoint.)

We will design a greedy approximation algorithm for this WD problem. To motivate the algorithm, we first consider two greedy algorithms that fail to achieve the target performance guarantee of $O(\sqrt{m})$.

Example 3.7 Suppose we sort the bids in decreasing order of value, and grant them greedily. In other words, we go through the bids one-by-one in sorted order, and we grant a bid if and only if all of its items are still available.

The following example is bad for this algorithm. There is a set S of m goods and $n = m+1$ players. Set $A_1 = S$ and $\alpha_1 = 1+\epsilon$ where $\epsilon > 0$ is arbitrarily small. For $i \in \{2, 3, \ldots, m+1\}$, set $\alpha_i = 1$ and A_i equal to the $(i-1)$th good of S. Our greedy algorithm grants the first bid and achieves a surplus of $1+\epsilon$; the optimal solution grants the rest of the bids and achieves a surplus of m. Thus this algorithm is no better than an m-approximation for the WD problem.

The greedy algorithm in Example 3.7 performs poorly because it fails to account for the fact that a big bid (i.e., a bid for many items) can block a large number of small bids that each have almost the same value as the big one. A natural way to fix this problem is to somehow normalize the value of a bid according to the number of items that it requires. This motivates our second greedy algorithm.

Example 3.8 Suppose we instead sort the bids in decreasing order of $\alpha_i/|A_i|$ (value-per-good) and grant bids greedily. This algorithm certainly returns the optimal solution for the input in Example 3.7. What is its performance in general?

Consider the following example: a set S of m goods, one player with $A_1 = S$ and $\alpha_1 = m - \epsilon$, and a second player with $A_2 = \{1\}$ and $\alpha_2 = 1$. The above greedy algorithm grants the second bid. The optimal solution grants the first bid. Thus the greedy algorithm is no better than an m-approximation algorithm for maximizing the surplus.

The greedy algorithm in Example 3.8 performs poorly because it undervalues large bids that primarily comprise items for which there is no contention.

Our final algorithm, the *LOS algorithm* due to Lehmann, O'Callaghan, and Shoham [5], interpolates between the greedy algorithms of Examples 3.7 and 3.8 and considers bids in decreasing order of $\alpha_i/\sqrt{|A_i|}$ (see Figure 1).

Exercise 3.9 Modify Examples 3.7 and 3.8 to obtain two different examples showing that the LOS algorithm is no better than a $\sqrt{m}$-approximation algorithm for the WD problem.

Perhaps surprisingly, this simple modification is enough to obtain an essentially best-possible approximation ratio (recall Corollary 3.6).

Theorem 3.10 ([5]) *The LOS algorithm is a $\sqrt{m}$-approximation algorithm for the WD problem with single-minded bidders.*

Input: A set S of m goods, (truthful) bids $(A_1, \alpha_1), \ldots, (A_n, \alpha_n)$.

1. Reindex the bids so that

$$\frac{\alpha_1}{\sqrt{|A_1|}} \geq \frac{\alpha_2}{\sqrt{|A_2|}} \geq \cdots \frac{\alpha_n}{\sqrt{|A_n|}}. \tag{9}$$

2. For $i = 1, 2, \ldots, n$: if no items of A_i have already been assigned to a previous player, set $T_i = A_i$; otherwise, set $T_i = \emptyset$.

Figure 1: The LOS approximate winner-determination algorithm.

Proof: Fix a set S of m goods and bids $(A_1, \alpha_1), \ldots, (A_n, \alpha_n)$. Let $X \subseteq \{1, 2, \ldots, n\}$ denote the indices of the bids granted by the LOS greedy algorithm, and X^* those of an optimal set of bids. We need to show that

$$\sum_{i^* \in X^*} \alpha_{i^*} \leq \sqrt{m} \cdot \sum_{i \in X} \alpha_i. \tag{10}$$

Our proof approach is a natural one for analyzing a greedy algorithm: we use the greedy criterion (9) to establish a "local bound" between "pieces" of the greedy and optimal solutions, and then combine these local bounds into the global bound (10).

We next make a simple but crucial definition. We say that a bid $i \in X$ *blocks* a bid $i^* \in X^*$ if $A_i \cap A_{i^*} \neq \emptyset$. We allow $i = i^*$ in this definition. Note that if i blocks i^* and $i \neq i^*$, then the bids A_i and A_{i^*} cannot both be granted; the greedy and optimal algorithms made different decisions as to how to resolve this conflict. For a bid $i \in X$, let $F_i \subseteq X^*$ denote the bids of X^* *first* blocked by i (i.e., $i^* \in X^*$ is placed in F_i if and only if i is the first bid in the greedy ordering that blocks i^*).

Two key points. First, we can already describe our "local bound" relating pieces of the optimal and greedy solutions. Suppose $i^* \in F_i$—the bid $i^* \in X^*$ is first blocked by $i \in X$. Then at the time the greedy algorithm chose to grant the bid i, the bid i^* was not yet blocked and was a viable alternative; by (9), we must have

$$\frac{\alpha_i}{\sqrt{|A_i|}} \geq \frac{\alpha_{i^*}}{\sqrt{|A_{i^*}|}} \tag{11}$$

whenever $i^* \in F_i$. The second key point is that each optimal bid $i^* \in X^*$ lies in precisely one set F_i. (Each bid $i^* \in X^*$ must be blocked by at least one bid of X—possibly by itself—since i^* would only by passed over by the greedy algorithm if it was blocked by some previously granted bid.) Thus the F_i's are a partition of X^*; in particular,

$$\sum_{i^* \in X^*} \alpha_{i^*} = \sum_{i \in X} \sum_{i^* \in F_i} \alpha_{i^*}. \tag{12}$$

This fact allows us to consider each bid $i \in X$ separately and then combine the results to obtain the global bound (10).

Now fix a bid $i \in X$. Summing over all $i^* \in F_i$ in (11), we have

$$\sum_{i^* \in F_i} \alpha_{i^*} \leq \frac{\alpha_i}{\sqrt{|A_i|}} \left(\sum_{i^* \in F_i} \sqrt{|A_{i^*}|} \right). \tag{13}$$

(Compare to (10).) The key question is: how big can the expression in parentheses on the RHS of (13) be? First, since all bids of F_i were simultaneously granted by the optimal solution, they must be disjoint and hence

$$\sum_{i^* \in F_i} |A_{i^*}| \leq m.$$

The worst case is that this inequality holds with equality. How would we then partition S among the $|F_i|$ bids of F_i to maximize $\sum_{i^* \in F_i} \sqrt{|A_{i^*}|}$? The answer is that we would spread the goods out equally ($m/|F_i|$ goods in each set). Formally this follows from the Cauchy-Schwarz inequality or from the concavity of the square-root function; it should also be easy to convince yourself of this fact with simple examples (e.g. the $|F_i| = 2$ case). These facts and (13) give

$$\sum_{i^* \in F_i} \alpha_{i^*} \leq \frac{\alpha_i}{\sqrt{|A_i|}} \left(\sum_{i^* \in F_i} \sqrt{\frac{m}{|F_i|}} \right) = \sqrt{m} \cdot \frac{\alpha_i}{\sqrt{|A_i|}} \sqrt{|F_i|}. \tag{14}$$

Finally, since the bid i blocks all of the bids of F_i, and bids of F_i are disjoint, in the worst case each item of A_i blocks a distinct bid of F_i (cf., Example 3.7). Thus $|F_i| \leq |A_i|$, which implies

$$\sum_{i^* \in F_i} \alpha_{i^*} \leq \sqrt{m} \cdot \alpha_i;$$

summing over all $i \in X$ and applying (12) completes the proof of (10). ■

Exercise 3.11 Suppose we modify the LOS algorithm to grant bids greedily in decreasing order of $\alpha_i/|A_i|^p$, where $p \in [0, 1]$ is a parameter. What is the approximation ratio of this algorithm, as a function of p?

3.3 A Truthful Payment Scheme

Now that we've designed a best-possible approximate WD algorithm (subject to the constraint of poly-time computation), we next aim to extend it to a truthful mechanism by charging suitable prices. In particular, recall that the LOS algorithm assumes that its input is a set of truthful bids; to justify this assumption, we seek prices that result in a strategyproof mechanism. (Otherwise the algorithm is optimizing using the wrong input, so its approximation guarantee is meaningless.)

A natural idea is to plug the LOS WD algorithm into step (2) of the VCG mechanism. In other words, first all players report their set A_i and value α_i, then we determine an allocation using the LOS algorithm, and then we charge player i a price equal to the monetary damage it causes the other players. Note that this is a poly-time mechanism. But is it truthful?

Example 3.12 Consider the following modification to Example 3.7. The first player has the set $A_1 = S$ and value $\sqrt{m} + \epsilon$. For $i = 2, 3, \ldots, m+1$, the ith player wants only the $(i-1)$th item and has value $\alpha_i = 1$.

If all players bid truthfully, then the LOS algorithm will grant only the first player's bid. But if we delete the first player's bid, then all of the other players' bids will be granted by the LOS algorithm. Thus the monetary damage caused by the first player to the rest equals m. But then the price charged to the first player by the VCG mechanism is m, even though its bid was only $\approx \sqrt{m}$, and this player winds up with negative utility! Thus the VCG mechanism together with the LOS algorithm is not truthful (e.g. the first player could obtain zero utility by bidding a value of 0), and is not even individually rational in the sense of Proposition 1.7.

In fact, the VCG mechanism is incompatible with approximate WD algorithms in a quite general sense; see Nisan and Ronen [8] for a detailed study of this issue.

The moral of Example 3.12 is that if we want to extend the LOS algorithm to a truthful mechanism, then we have to carefully design a pricing scheme that is tailored to the algorithm. The solution to this non-trivial problem follows.

The high-level idea of the LOS pricing scheme is to charge prices that are "Vickrey-like", in the sense that a winner i should pay according to a suitable function of the highest-value bid that i's bid blocks. This motivates a key definition.

Definition 3.13 Suppose bid i was granted by the LOS algorithm while bid j was denied. The bid i *uniquely blocks* the bid j if, after deleting the bid i from the input, the LOS algorithm grants the bid j.

We will use the terminology *u-blocks* as shorthand for "uniquely blocks". Definition 3.13 is somewhat subtle. We give a simple example, and encourage the reader to explore more complicated ones.

Example 3.14 Figure 2 shows a rough picture of four bids. The bids are numbered according to the LOS greedy ordering. Overlap between two circles is meant to indicate that the two bids share at least one item. Given the full input, the LOS algorithm will grant the first two bids and deny the last two. If the first bid is deleted, the LOS algorithm will grant the second and fourth bids. Thus the first bid u-blocks the fourth bid, but it does not u-block the third bid.

Exercise 3.15

(a) Show that the terminology "u-block" is somewhat misleading in the following sense: a bid (B_i, b_i) can u-block a bid (B_j, b_j) even if B_i and B_j are disjoint.

(b) On the other hand, show that if (B_j, b_j) is the *first* bid in the LOS ordering that is u-blocked by (B_i, b_i), then $B_i \cap B_j \neq \emptyset$.

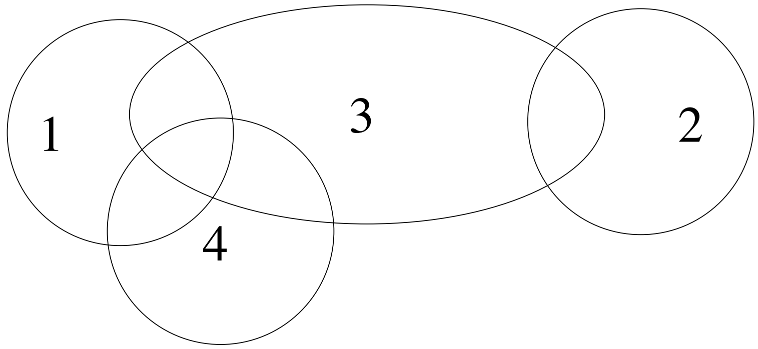

Figure 2: Illustration of Definition 3.13 (u-blocking).

The idea of the LOS pricing scheme is to charge a winning bidder according to the highest-value bid that it u-blocks. Here "highest-value" should be suitably normalized by bid size, to reflect the way the LOS algorithm chooses its ordering. Precisely, the LOS prices are as follows.

- If the bidder i loses, or if its bid wins but u-blocks no other bid, then $p_i = 0$.
- Otherwise, suppose i's bid is (B_i, b_i), and let (B_j, b_j) be the first bid in the LOS greedy ordering that i's bid u-blocks. Set

$$p_i = \frac{b_j}{\sqrt{|B_j|}} \cdot \sqrt{|B_i|}. \tag{15}$$

By the *LOS mechanism*, we mean the CA that uses the WD algorithm of Subsection 3.2 followed by the above charging scheme.

Individual rationality is almost immediate.

Proposition 3.16 *Truthtelling bidders always obtain nonnegative utility in the LOS mechanism.*

Proof: We need to show that the price p_i charged to a winning bidder i is at most its bid b_i. Let (B_j, b_j) be the first bid that (B_i, b_i) u-blocks (if there is no such bid, then $p_i = 0$ and there's nothing to prove). Since (B_j, b_j) must follow (B_i, b_i) in the LOS ordering,

$$\frac{b_i}{\sqrt{|B_i|}} \geq \frac{b_j}{\sqrt{|B_j|}};$$

rearranging gives $b_i \geq p_i$, as desired. ■

Strategyproofness is much less obvious.

Theorem 3.17 *The LOS mechanism is strategyproof.*

Again, we leave it to the reader to investigate the extent to which the LOS mechanism is strongly truthful in the sense of Proposition 1.6.

Our first step in proving Theorem 3.17 is to show that bidders have no incentive to lie about their desired sets (the A_i's).

Lemma 3.18 *If a player i can benefit in the LOS mechanism from a false bid (B_i, b_i), then it can benefit from such a bid in which $B_i = A_i$.*

Proof: Suppose there is a player i and a set of bids $\{(B_j, b_j)\}_{j \neq i}$ for the other $n-1$ players such that i obtains strictly greater utility from falsely bidding (B_i, b_i) than from truthfully bidding (A_i, α_i). By Proposition 3.16, this can only occur if the LOS mechanism grants the bid (B_i, b_i). We aim to show that the false bid (A_i, b_i) also leads to greater utility than the bid (A_i, α_i).

First note that in the false bid (B_i, b_i), we must have $B_i \supseteq A_i$: if B_i is missing any items from A_i, then the LOS mechanism will never produce an outcome in which i has strictly positive utility. (And by Proposition 3.16, a truthful bid always leads to nonnegative utility.) So suppose B_i contains A_i and that the LOS mechanism grants the bid (B_i, b_i); we can complete the proof by showing that the LOS mechanism would have also granted the bid (A_i, b_i) and would have only charged player i a smaller price.

The first part of the above statement is easy to see: since $A_i \subseteq B_i$, the bid (A_i, b_i) would only be considered earlier in the greedy LOS ordering (9) and would therefore be granted. For the second part, recall from (15) that the price charged to player i by the LOS mechanism is $p_i = b_j\sqrt{|B_i|}/\sqrt{|B_j|}$, where j is the earliest bid u-blocked by i (if any). Bidding A_i instead of B_i affects this price in two ways. First, the second term on the RHS of (15) clearly only goes down. The second, trickier consequence is that the identity of the first u-blocked bid could change. So suppose the first bid u-blocked by the bid (B_i, b_i) is (B_j, b_j) and that by (A_i, b_i) is (B_k, b_k). (To rule out the possibility that there is no u-blocked bid, add an imaginary bid for all of the items that has zero value.) The final key claim, which we leave as an exercise, is that (B_k, b_k) can only follow (B_j, b_j) in the greedy LOS ordering. This implies that bidding A_i instead of B_i can only decrease the first term on the RHS of (15), and completes the proof. ■

Exercise 3.19 Complete the proof of Lemma 3.18: assume that $B_i \supseteq A_i$ and show that if (B_j, b_j) and (B_k, b_k) are the first bids u-blocked by the bids (B_i, b_i) and (A_i, b_i), respectively, then (B_k, b_k) can only follow (B_j, b_j) in the greedy LOS ordering (9).
[See also the proof of Theorem 3.17 below for a similar argument.]

We now complete the proof of Theorem 3.17.

Proof of Theorem 3.17: As in the proof of Lemma 3.18, assume for contradiction that there is a player i and a set of bids $\{(B_j, b_j)\}_{j \neq i}$ for the other $n-1$ players such that i obtains strictly greater utility from falsely bidding (B_i, b_i) than from truthfully bidding (A_i, α_i). By Lemma 3.18, we can assume that $B_i = A_i$. Let $\mathcal{B}_{-i}$ denote the set $\{(B_j, b_j)\}_{j \neq i}$ of other players' bids; $\mathcal{B}_T$ the set $\mathcal{B}_{-i} \cup \{(A_i, \alpha_i)\}$; and $\mathcal{B}_F$ the set $\mathcal{B}_{-i} \cup \{(A_i, b_i)\}$. By Proposition 3.16, we can assume that the LOS mechanism granted the bid (A_i, b_i) given the input $\mathcal{B}_F$.

There are two cases. We consider only the case where $b_i < \alpha_i$, and leave the other case as an exercise. We can assume that the bid (A_i, b_i) was granted. Since $\alpha_i > b_i$, the bid (A_i, α_i) would have only been considered earlier in the LOS ordering and thus would also have been granted. Suppose that (B_j, b_j) is the first bid u-blocked by the false bid (A_i, b_i). We can complete the proof by showing that (A_i, α_i) does not u-block any bid earlier than (B_j, b_j), as then the price (15) charged by the LOS mechanism on input $\mathcal{B}_T$ for the bid (A_i, α_i) is at most that for the bid (A_i, b_i) on the input $\mathcal{B}_F$.

Suppose for contradiction that the first bid (B_k, b_k) that (A_i, α_i) u-blocks precedes (B_j, b_j) in the LOS ordering. By the definition of u-blocking, removing the bid (A_i, α_i) from $\mathcal{B}_T$ and rerunning the LOS algorithm on the input $\mathcal{B}_{-i}$ causes the bid (B_k, b_k) to be granted. A key observation is this: *if (A_i, b_i) follows (B_k, b_k) in the LOS ordering, (B_k, b_k) would also be granted by the LOS algorithm on the input $\mathcal{B}_F$*—this holds because the LOS algorithm makes identical decisions on the inputs $\mathcal{B}_{-i}$ and $\mathcal{B}_F$, until the point that the bid (A_i, b_i) is considered in the latter execution. Since Exercise 3.15(b) implies that A_i and B_k must have at least one item in common, and since the bid (A_i, b_i) is granted by the LOS algorithm given the input $\mathcal{B}_F$, this observation implies that (A_i, b_i) precedes (B_k, b_k) in the LOS ordering. But then (A_i, b_i) u-blocks (B_k, b_k), contradicting the assumption that (B_k, b_k) precedes the first bid (B_j, b_j) u-blocked by (A_i, b_i). ■

Exercise 3.20 Complete the proof of Theorem 3.17: show that if (b_i, A_i) is a winning bid and $b_i > \alpha_i$, then player i's utility would have been at least as large had it bid (α_i, A_i).

Exercise 3.21 Suppose we modify the LOS mechanism so that the price p_i charged for a winning bid (B_i, b_i) is given by (15), but where the bid (B_j, b_j) is defined as the first bid blocked by (B_i, b_i)—the first denied bid after (B_i, b_i) with $B_i \cap B_j \neq \emptyset$. Does this result in a strategyproof mechanism?

Exercise 3.22 Recall from Exercise 3.11 that the LOS WD algorithm can be extended to a family of greedy algorithms, parametrized by p. Can all of these WD algorithms be extended to truthful mechanisms via appropriate pricing schemes? What about for other classes of greedy criteria (e.g. ordering bids according to $\alpha_i/f(|A_i|)$, where f is a more general nondecreasing function of set size)?

3.4 Summary

This section studied the LOS CA for single-minded bidders. On the plus side, this is our first poly-time CA for valuations that can have some degree of complements or substitutes (in this case, a restricted form of complements). On the minus side, the valuations can have only a very restricted form and the CA guarantees only a relatively weak ($O(\sqrt{m})$) approximation of the maximum surplus. In terms of our guiding desiderata (P1)–(P4) from Subsection 1.3, the LOS CA achieves incentive compatibility (P1) and computational tractability (P4) while making serious concessions to economic efficiency (P2) and valuation generality (P3). We have already seen (Corollary 3.6) that the trade-off between economic efficiency and computational tractability is fundamental, even for single-minded bidders, and even ignoring incentive compatibility. The next section shows that even a weaker notion of CA tractability—poly-time communication and unbounded computation—leads to a fundamental trade-off, between economic efficiency and valuation generality.

4 Communication Complexity of CAs

Last section restricted attention to single-minded bidders in part to eliminate communication difficulties and focus on the computational complexity of winner determination. This section returns to general valuations — where all we know about each valuation v_i is that $v_i(\emptyset) = 0$ and that $v_i(T_1) \leq v_i(T_2)$ whenever $T_1 \subseteq T_2$ — and shines the spotlight squarely on communication issues.

Intuitively, since a general valuation has an exponential number of free parameters, we don't expect to achieve a reasonable allocation in all cases while examining only a polynomial number of them. To make this precise, we consider the following model of computation. (See [4] for an overview of the various standard models.) Players participate in a protocol, decided upon in advance; at each step of the protocol, one of the players transmits a bit, which is seen by all players. Crucially, the bit transmitted by a player can only depend on its own private information and the protocol history so far (i.e., who transmitted what). The *communication complexity* of a protocol is the worst-case number of bits that are transmitted (over all possible private inputs of the players).

The key point to take away from this definition is how powerful the model of computation is: in addition to dispensing with any incentive constraints (which we will do for this entire section), *unlimited computation* by the players is permitted. While the point of this model is lower bounds (which are only more compelling in such an unrealistically strong model), let's develop some intuition by examining some positive results.

First, observe that winner determination with single-minded bidders is trivially solvable with a polynomial amount of communication. The following protocol works: (1) each player broadcasts their private set and value in some predetermined order (recall we ignore incentive constraints); and (2) each player uses these to compute an optimal solution in a consistent way (this is an NP-hard problem, but recall we allow unbounded computation).

Second, the LOS algorithm can be used to achieve a non-trivial approximation guarantee with polynomial communication even for *general* valuations in this model of computation. The idea is to conceptually treat a single player with a general valuation v_i as 2^m different single-minded players — one single-minded player for each bundle $T \subseteq S$, with inherited valuation $v_i(T)$. To prevent different "sub-players" corresponding to a single original player from simultaneously getting their bundles granted, we add one "dummy good" for each original player i. We then supplement the desired set of each of i's sub-players with this dummy good. This ensures that every feasible allocation with the sub-players and the dummy goods maps naturally to a feasible allocation of the original instance with the same surplus.

We have shown how to reduce surplus maximization with n players with general valuations and m goods to surplus maximization with $n2^m$ single-minded players and $m+n$ goods. Solving the latter "single-minded instance" by brute-force (as above) would require communication exponential in one of the original parameters of interest (namely, m). Running the LOS algorithm directly on the single-minded instance suffers the same problem.

We can simulate the decisions that the LOS algorithm would make on the single-minded instance, using only polynomial (in n and m) communication, as follows. We define a protocol that works directly on the original instance (with n players and m goods). The protocol proceeds in rounds. All players are initially active and all goods are initially unallocated. In each round, each active player i broadcasts the bundle T_i^* of unallocated goods that maximizes $v_i(T_i)/\sqrt{|T_i|}$. (Solving this maximization problem might require exponential time by the player, but remember this is permitted.) All players see all proposed bundles, and the one that maximizes $v_i(T_i^*)/\sqrt{|T_i^*|}$ over active players i is understood by all of the players to be allocated. The winning player i^* deactivates itself and the goods in its bundle $T_{i^*}^*$ are understood by all players to now be allocated. The protocol terminates once all players are inactive. Intuitively, each round of the protocol is executing a "two-stage tournament" to identify the bundle that would next be selected by the LOS algorithm on the induced single-minded instance — in the first stage, each original player runs a tournament to elect the most viable candidate from its 2^m induced single-minded players (this can be done privately, without any communication), and the second round elects a final winner from the polynomially many candidates that survive the first stage.

Exercise 4.1 Prove that the allocation decisions made by the above protocol for the original instance are isomorphic to those that the LOS winner determination algorithm would make on the induced single-minded instance, and therefore it achieves an $O(\sqrt{m})$-approximation of the surplus.

The main result in this section is a matching lower bound.

Theorem 4.2 ([6]) *For every $\epsilon > 0$, there is no polynomial-communication, $O(m^{(1/2)-\epsilon})$-approximation for the general winner determination problem.*

This lower bound is "unconditional", in that it doesn't depend on any complexity-theoretic assumptions like $P \neq NP$. It can be extended to cover randomized and nondeterministic protocols, and similar proof techniques also yield (sometimes weaker) lower bounds for various restricted classes of valuations. See [7, 10] for further details and references.

At the highest level, the proof of Theorem 4.2 is not unlike the familiar argument that comparison-based sorting requires $\Omega(n \log n)$ comparisons — an algorithm that employs only k comparisons generates at most 2^k distinct executions, and $n!$ different executions are needed to correctly distinguish the $n!$ ordinally distinct possible inputs. (Recall $\log_2 n! = \Theta(n \log n)$.) The proof of Theorem 4.2 needs two additional ideas. First, the structure of the private information implies that sets of inputs that generate identical protocol transcripts satisfy a natural closure property. Second, to prove the strong approximation lower bound of $\Omega(m^{(1/2)-\epsilon})$ we require some neat combinatorics to generate winner determination instances that admit either a high-surplus feasible solution or only very low-surplus solutions.

The first point is simple. Consider a protocol and let X_i denote the set of possible private inputs of player i (e.g., possible valuations). Suppose there are two inputs $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ for which the communication transcripts of the protocol (i.e., who sent what bits when) are identical. Now consider the "mixed" input $(y_1, x_2, x_3, \ldots, x_n)$. By induction on the rounds of the protocol: (1) player 1 cannot distinguish between the inputs $(y_1, x_2, x_3, \ldots, x_n)$ and $(y_1, y_2, y_3, \ldots, y_n)$; and (2) the other players cannot distinguish between the inputs $(x_1, x_2, x_3, \ldots, x_n)$ and $(y_1, x_2, x_3, \ldots, x_n)$. As part of this induction, we see that the communication transcript of the protocol on the input $(y_1, x_2, x_3, \ldots, x_n)$ matches that of $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$. Similarly, all "mixed versions" of $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ generate identical communication transcripts. This implies that a set of inputs with a common communication transcript form a *box*, meaning a subset A of $X_1 \times \cdots \cdots X_n$ that arises as a product: $A = A_1 \times \cdots \times A_n$ for some $A_i \subseteq X_i$ for each i.

Lemma 4.3 *Every protocol partitions the set $X = X_1 \times \cdots \times X_n$ of possible inputs into boxes over which its communication transcript is invariant.*

For the winner determination problem, a protocol with communication complexity k partitions the set of valuations into at most 2^k boxes, and in each box executes identically — in particular, a common allocation is produced for all inputs in the same box. The heart of the proof of Theorem 4.2 is to show that if k is too small (i.e., polynomial), then very different-looking inputs wind up in a common box, and no common allocation can be simultaneously near-optimal for both of them.

To construct a useful family of "different-looking" valuations, fix a set S of m goods and a set of $n = \Theta(m^{(1/2)-\epsilon})$ players. We first consider the following thought experiment. Make t different copies of the goods S, called $S^1, \ldots, S^t$, where t is a parameter we choose below. Randomly partition each S^j into n classes, one per player (i.e., assign each good of S^j independently and uniformly at random to one of the classes $S^j_1, \ldots, S^j_n$). Obviously, two differ classes in the same copy S^j contain disjoint subsets of the original set S of goods. What about two classes S^j_i, S^ℓ_h belonging to different copies ($j \neq \ell$)? For each original good of S, there is a $1/n^2$ probability that it is assigned to both S^j_i and S^ℓ_h. Thus, for fixed h, i, and $j \neq \ell$, the probability that S^j_i and S^ℓ_h wind up disjoint is $(1 - 1/n^2)^m < e^{-m/n^2}$. Note that under our assumption that $n = \Theta(m^{(1/2)-\epsilon})$, this probability is exponentially small. Indeed, by a Union Bound, the probability that there is *any pair* of sets S^j_i, S^ℓ_h with $j \neq \ell$ and no good of S in common is less than $t^2 n^2 e^{-m/n^2}$. Thus, even when

$$t = \frac{1}{n} e^{m/2n^2}, \tag{16}$$

there is a positive probability that every pair S^j_i, S^ℓ_h of classes with $j \neq \ell$ overlaps. Ergo, such a collection of t partitions of the goods S exists; we fix one $\{S^j_i\}$ arbitrarily for the rest of the proof.

Why is this construction useful? To gain intuition, suppose each bidder i was single-minded and wanted the bundle S^1_i, with value 1. Then we can allocate all desired bundles to all bidders without conflict and enjoy surplus $n = \Theta(m^{(1/2)-\epsilon})$. If, on the other hand, each bidder wants a bundle that corresponds to a different copy of the goods, we can only obtain surplus 1 (recall every pair of classes from different partitions has at least one good of S in common). Thus this collection of t highly overlapping partitions of S generates winner determination instances with both very high optimal surplus and very low optimal surplus.

We now give the general argument and prove Theorem 4.2. We first describe the set of valuations that we use. Let $B_i \subseteq \{0,1\}^t$ be a bit string of length t; associate these t bits with the t partitions of S above. Interpret the ones of B_i as the partitions in which player i is interested, and the zeros as the partitions in which it is uninterested. The string B_i induces a valuation as follows: for every copy S^j in which i is interested, player i has value 1 for the bundle S_i^j. The player also has value 1 for supersets of such bundles, and value 0 for everything else. Let X_i denote the set of 2^t valuations of this form. The set $X = X_1 \times \cdots \times X_n$ of inputs induces a family of winner determination problems.

Consider an input of X, which we can uniquely associate with bit strings $B_1, \ldots, B_n$. Call an instance *good* if there is an index h such that, for every player i, the hth bit of B_i is 1 (i.e., all players are interested in the hth partition). As above, a good instance admits a feasible solution with surplus $n = \Theta(m^{(1/2)-\epsilon})$, in which each player gets its bundle corresponding to the hth partition. At the other extreme, call an instance *bad* if there is at most one player interested in each partition (i.e., the sets of indices for the ones in $B_1, \ldots, B_n$ are mutually disjoint). Since all pairs of bundles drawn from different partitions intersect, the maximum-possible surplus in a bad instance is 1. (Of course, there are plenty of instances that are neither good nor bad.)

Finally, consider a k-bit protocol that achieves a better-than-n approximation for every winner determination problem in X. By Lemma 4.3, this protocol partitions X into at most 2^k boxes over which the protocol has constant behavior (and in particular, a constant output). By the definition of good and bad instances, and the assumption that the protocol is better than an n-approximation algorithm, good and bad instances cannot intermingle in a common box.

Crucially, this restricts the number of bad instances that a single box can contain. To see why, consider a box $A = A_1 \times \cdots \times A_n$ of X (recall Lemma 4.3) that contains no good instances. We claim that for each partition S^j of the goods, there is a "totally uninterested" player i — a player i who, across all of its valuations in A_i, never wants its bundle S_i^j from the jth partition. For otherwise, there is a partition S^j and, for each player i, a valuation $v_i \in A_i$ such that, when i has this valuation, it would happily accept its bundle from the jth partition. But then the input $v_1, \ldots, v_n$ belongs to this box (by the closure property of boxes) and, by definition, is a good instance. So the claim is true — but why does it imply an upper bound on the bad instance population of a box with no good instances? The *total* number of bad instances is precisely $(n+1)^t$, with each arising uniquely by choosing, for each of the t partitions, which (if any) one of the n players is interested in it. Within a box with no good instance, each bad instance arises a choice, one per partition, of which (if any) of the $n-1$ players *other than the necessarily present totally uninterested one*, is interested in it. This gives an upper bound of n^t on the number of bad instances per box.

Wrapping up, the $(n+1)^t$ bad instances must be distributed across at least $(n+1)^t/n^t =$

$(1+\frac{1}{n})^t$ different boxes. This implies that the communication complexity k of the protocol satisfies $2^k \geq (1+\frac{1}{n})^t$; taking logs and using that $\log(1+x) \approx x$ for small $x > 0$, we find that $k \geq t/n$. Since t is exponential in m (recall (16)), so is k. Summarizing, then: *every protocol with approximation factor* $o(\min\{n, m^{(1/2)-\epsilon}\})$ *uses exponential communication.*

References

[1] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.

[2] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[3] J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[4] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1996.

[5] D. Lehmann, L. I. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002. Preliminary version in *EC '99*.

[6] N. Nisan. The communication complexity of approximate set packing. In *Proceedings of the 29th Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 2380 of *Lecture Notes in Computer Science*, pages 868–875, 2002.

[7] N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9, pages 209–241. Cambridge University Press, 2007.

[8] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research*, 29:19–47, 2007.

[9] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1):1–54, 2002.

[10] I. Segal. The communication requirements of combinatorial allocation problems. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, chapter 11. MIT Press, 2006.

[11] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.