# CS364B: Frontiers in Mechanism Design Lecture #11: Undominated Implementations and the Shrinking Auction*

Tim Roughgarden†

February 12, 2014

## 1 Introduction to Part III

Recall the three properties we've been focused on so far.

1. **(Incentive guarantee.)** DSIC or EPIC.
2. **(Performance guarantee.)** Assuming honest behavior, the auction should output an allocation with (approximately) optimal welfare.
3. **(Tractability guarantee.)** The auction should be "simple" or at least terminate in a reasonable amount of time.

Part I of the course focused on special cases where we can achieve all three properties — gross substitutes (GS) valuations and special cases thereof. In Part II of the course, we focused on more general valuation classes in which the exact version of the second property is already incompatible with the third (assuming $P \neq NP$). We focused further on special cases where, ignoring incentive constraints, a constant-factor approximation to the optimal welfare can be computed in polynomial time. We've seem some pleasing positive results: when there is a logarithmic supply of every item or with coverage valuations, we designed MIDR allocation rules (and hence DSIC mechanisms) that have best-possible approximation guarantees (assuming $P \neq NP$), albeit through quite complex designs.[1] For general submod-



†Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.

[1]Let's recall the technical fine print. The result for logarithmic supply assumes that each bidder only wants one copy of each item and that the valuations are given as black boxes that support demand queries. The approximation guarantee is $(1-\epsilon)$ provided the supply is at least $c\epsilon^{-2}\log m$ for some (modest) constant $c$. The approximation guarantee for coverage valuations is $1-\frac{1}{e} \approx 0.63$. This guarantee extends to valuations that are convex combinations of gross substitutes valuations, assuming that such valuations are given as black boxes that support a randomized version of a value oracle.

ular valuations (which subsume coverage valuations), the situation is different and there are impossibility results that rule out analogous DSIC approximation guarantees. Both Parts I and II adopt the first and third properties above as hard constraints, and then explore what is possible for the second property.

The complexity and limited reach of polynomial-time DSIC mechanisms motivate compromising on the incentive guarantee, in the hopes of designing simpler mechanisms with stronger welfare guarantees. This is the focus of Part III of the course (Lectures #11–13).

What is a principled way to relax the DSIC incentive guarantee? The traditional answer, and the one we study is Lectures #12 and #13, is to require "Bayes-Nash incentive compatibility." We'll define this precisely next lecture; for now we only mention that the definition requires a common prior over bidders' valuations.

But first, in this lecture, we take a detour into a compellingly mild relaxation of the DSIC definition, for which few positive or negative results are currently known. (Thus, this is an intriguing direction for future research.) To explain the rough idea, recall the DSIC condition guarantees that each player has a "foolproof" (i.e., dominant) strategy, and the assumption is that players play dominant strategies when they exist. In this lecture, we design a mechanism without any dominant strategies, and assume only that bidders don't play "stupid" (i.e., dominated) strategies. We lose the ability to predict exactly what players will do, but we'll identify a mechanism in which the welfare is near-optimal no matter what (non-stupid) strategies the players choose.

# 2 Single-Valued, Multi-Minded Bidders

This lecture, we consider bidders who have a private valuation $v_i \in \mathcal{R}$ for getting any bundle from a desired collection $\{A_{i\ell}\}$ of bundles.

**Scenario #9:**

- A set $U$ of $m$ non-identical items.
- Each bidder $i$ has a private value $v_i$ for each of a collection $\{A_{i\ell}\}$ of private bundles. Thus, the full valuation is

$$v_i(S) = \begin{cases} v_i & \text{if } S \supseteq A_{i\ell} \text{ for some } A_{i\ell} \\ 0 & \text{otherwise.} \end{cases}$$

Importantly, we assume that both the value $v_i$ and the sets $\{A_{i\ell}\}$ are private to bidder $i$. This is reminiscent of unit-demand valuations, except with bundles instead of individual items, and with a common value for all desired bundles. This valuation class is clearly restricted, though one can imagine scenarios where it is relevant — for example, if a firm has value $v_i$ for getting a project completed, and the $A_{i\ell}$'s are teams of workers that collectively possess the skills needed for the project. Another example is where $U$ represents the edges of a network, a bidder has a private origin $s_i$ and destination $t_i$, the bidder has value $v_i$ for connecting its origin to its destination, and the $A_{i\ell}$'s are the $s_i$-$t_i$ paths of the network.

Primarily, though, we focus on this scenario to illustrate an interesting relaxation of the DSIC condition.

We assume throughout this lecture that all private bundles $A_{i\ell}$ have size at most a publicly known parameter $d$. The underlying welfare-maximization problem is $NP$-hard as long as $d \geq 3$. (It can be solved in polynomial time for $d = 2$, see Exercises.) Both the mechanism description and its performance will depend on $d$. For example, you might want to keep in mind the case where $d = 6$ and most bundles range in size from 1 to 4. We also assume that all of the private values $v_i$ for desired sets are bounded below by some publicly known constant (say 1).

# 3 Some Special Cases

To get a feel for Scenario #9, let's examine some special cases. First suppose that all desired bundles $\{A_{i\ell}\}$ are publicly known, and only the values $v_i$ for desired bundles are private. We then have a single-parameter environment (CS364A, Lecture #4). Consider the allocation rule that uses the greedy algorithm:

1. Go through the bidders one-by-one, from highest to lowest bid.

2. Upon reaching bidder $i$, if for some desired bundle $A_{i\ell}$ all items of $A_{i\ell}$ are still available, award bidder $i$ an arbitrary such bundle $A_{i\ell}$.

This greedy algorithm clearly runs in polynomial time. Because each accepted bidder can only "block" $d$ other bidders, the algorithm outputs an allocation with welfare at least $1/d$ times the maximum possible (see Exercises for details). A reduction from the Independent Set problem in $d$-regular graphs shows that no polynomial-time algorithm has approximation ratio $o(d/\log d)$, assuming $P \neq NP$ [2]. The allocation rule is monotone (Exercises), and hence Myerson's Lemma (CS364A, Lecture #3) can be used to extend it to a DSIC mechanism.

For a second special case, suppose that each bidder $i$ has only one desired bundle $A_i$ (i.e., is "single-minded") but that this bundle is private. We again consider the rule that computes an allocation using the greedy algorithm above. The environment is no longer single-parameter, so we can't use Myerson's Lemma to prove that the rule yields a DSIC mechanism. Nevertheless, one can prove directly that natural payments turn the greedy allocation rule into a DSIC mechanism (see Exercises) [3]. Of course, the rule continues to provide a $\frac{1}{d}$-approximation of the optimal welfare.

# 4 The Shrinking Mechanism

## 4.1 Description of Mechanism

We now describe an indirect mechanism, the "shrinking mechanism," for Scenario #9. After describing the mechanisms, we discuss some of its properties and go through a detailed example. We then make precise our behavioral assumptions, and prove an approximation guarantee for the mechanism under these assumptions.

**Shrinking Mechanism** [1]

1. Initialize $b_i = 1$ for every bidder $i$.

   [The current bid of bidder $i$; only goes up as the auction proceeds. Recall we assume $v_i \geq 1$ for every $i$.]

2. Initialize $S_i = U$ for every bidder $i$.

   [The items that $i$ currently wants; only shrinks as the auction proceeds.]

3. Initialize $\textsc{old} = \emptyset$, $\textsc{losers} = \emptyset$.

   [Two sets of bidders.]

4. While there is a bidder $i \notin \textsc{old} \cup \textsc{losers}$:

   [We simulate the greedy algorithm on the bidders not in $\textsc{losers}$.]

   (a) Iterate through the bidders $i \notin \textsc{losers}$, from highest to lowest bid $b_i$:

      i. If $|S_i| \leq d$ and $S_i$ is feasible — i.e., disjoint from $\cup_{h \in \textsc{new}} S_h$:

         A. Add $i$ to $\textsc{new}$.

      ii. Else ask bidder $i$ to choose between two options:

         A. **Option 1:** Shrink the set $S_i$ to some $T_i \subset S_i$ with $|T_i| \leq d$ and $T_i$ disjoint from $\cup_{h \in \textsc{new}} S_h$. In this case:
            - $S_i \leftarrow T_i$.
            - Add $i$ to $\textsc{new}$.

         B. **Option 2:** Pass ($S_i$ remains the same).

   (b) If $\sum_{i \in \textsc{new}} b_i > \sum_{i \in \textsc{old}} b_i$:
      [Ensure monotonicity.]

      i. $\textsc{old} \leftarrow \textsc{new}$.

   (c) For each $i \notin \textsc{old} \cup \textsc{losers}$, ask bidder $i$ to choose between two options:

      i. **Option 1:** Bid increase: $b_i \leftarrow 2b_i$.

      ii. **Option 2:** Drop out: add $i$ to $\textsc{losers}$.

5. Final allocation: $S_i$ to each bidder $i \in \textsc{old}$, $\emptyset$ to all other bidders.

6. Final payments: $b_i$ to each bidder $i \in \textsc{old}$, 0 to all other bidders.

## 4.2 Discussion

Here are several comments to aid understanding the auction:

1. The point of the main while loop (line 4) is to run the greedy algorithm on the bidders that haven't yet dropped out (i.e., $i \notin$ LOSERS) with respect to the current bids $\mathbf{b}$ and the current sets $\{S_i\}_{i \notin \text{LOSERS}}$. This isn't quite what happens, since the auction skips over all bidders that haven't shrunk their sets set (i.e., $S_i = U$). This is because the greedy algorithm only has a good approximation guarantee when all bundles have size at most $d$. Thus line 4 runs the greedy algorithm on the set of bidders $i \notin$ LOSERS that have shrunk their set (equivalently, that have $|S_i| \leq d$). This property is used in Lemma 5.4.

2. The allocation OLD is always a feasible allocation.

3. Line 4b ensures that the sum of the bids in the computed allocation can only go up each iteration. (Omitting this line can destroy this monotonicity; see the Exercises.)

   This monotonicity property is used in the proof of Lemma 5.4, though step 4b does complicate the proof of Lemma 5.2.

4. From a player's perspective, it is not easy to figure out how play in the shrinking auction. For example, suppose that player $i$'s set $S_i$ currently has size 6 and is the disjoint union of three bundles that $i$ desires. In line 4(a)ii, should $i$ shrink its set or pass (and double its bid)? If it shrinks, which subset should it shrink $S_i$ to?

   For this reason, we cannot hope to have a sharp prediction about what happens in the shrinking action. Nevertheless, we prove in Section 5 that all "reasonable" outcomes of the shrinking auction have near-optimal welfare.

## 4.3 An Example

Next we run the shrinking auction on a concrete example.

**Example 4.1** Suppose there are $n = 4$ bidders and a set $U = \{1, 2, 3, 4\}$ of 4 goods. Suppose $d = 2$ and each bidder has only one desired bundle of two goods, as shown in Figure 1. The private values are $v_1 = 6$, $v_2 = 5$, $v_3 = 12$, and $v_4 = 15$. In what follows, we assume that the shrinking auction always breaks ties in favor of lower-indexed players.

Initially all bids $b_i$ are 1 and all sets $S_i$ are $U$. The auction asks the first bidder whether or not it wants to shrink (to a set of at most $d = 2$ items) or pass. Assume for the example that the bidder shrinks its set $S_1$ from $U$ to the bundle $\{1, 2\}$ of items that it actually wants. Intuitively, this is reasonable because the bidder is only giving up on items that it doesn't want under any circumstances. Next, the second bidder is asked whether or not it wants to shrink or pass. The only way the bidder can shrink and be feasible with the first bidder is to set $S_i$ to $\{3, 4\}$. This bundle has no value for the bidder (it only wants $\{2, 4\}$), so we assume that the second bidder passes instead. When the third bidder is asked to shrink or pass, it

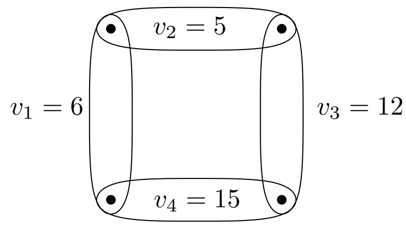


Figure 1: Example of an instance of the shrinking auction.

may as well shrink to the set $S_3 = \{3, 4\}$ of items that it actually wants. The fourth bidder will certainly choose to pass over shrinking (necessarily to $\emptyset$, since all items are spoken for). This completes the allocation phase of the first iteration, at which point NEW $= \{1, 3\}$. This allocation certainly has higher sum of bids than OLD $= \emptyset$, so we save it in OLD. Bidders 2 and 4 are asked to choose between dropping out and doubling their bids. Since their values both exceed 2, it makes sense for them to stay in the auction, now with $b_2 = b_4 = 2$.

In the second iteration of the while loop, the players are approached in the order $2, 4, 1, 3$. Bidder 2 is asked to either shrink its set (from $U$ to a set of at most $d = 2$ items) or pass; it makes sense for bidder 2 to shrink its set to the items $\{1, 3\}$ that it actually wants. Similarly, it makes sense for bidder 4 to shrink its set to $\{2, 4\}$. Bidders 1 and 3 will pass rather than shrink (to the empty set). This allocation phase concludes with NEW $= \{2, 4\}$ and OLD $= \{1, 3\}$. Since the sum of the bids in the former (namely, 4) is more than in the latter (2), the allocation NEW is saved. Since bidders 1 and 3 have values more than 2, both will presumably choose to double their bids rather than drop out.

In the third iteration of the while loop, bidders 1 and 3 will be chosen (assuming bidder 2 passes rather than shrinks, as it should), so NEW $= \{1, 3\}$ and OLD $= \{2, 4\}$. Note that both allocations have sum of bids 4. Since the new allocation is not strictly bigger than the old one, the new one discarded and the old one retained. Bidders 1 and 3 are again asked to either double their bids (to 4) or drop out. Since bidders 1 and 3 both have valuation bigger than 4, both will presumably agree to the bid doubling.

In the fourth iteration of the while loop, bidders 1 and 3 are chosen. Now the sum of bids in NEW (namely, 8) is more than in OLD (4), so the allocation $\{1, 3\}$ is retained. Bidders 2 and 4 are asked either double their bid (to 4) or drop out; since both have value more than 4, both should agree to the doubling.

In the fifth iteration, bidders 1 and 3 are chosen again (assuming 2 passes rather than shrinks, as it should) so bidders 2 and 4 are asked again to double their bids. This time, we expect bidder 2 to drop out rather than double its bid to 8 (more than its valuation of 5). Bidder 4 should agree to the bid doubling.

In the sixth iteration, the bidders not in LOSERS are $\{1, 3, 4\}$ and their current bids are $b_1 = b_3 = 4$ and $b_4 = 8$. Bidder 4 will be chosen first, and bidders 1 and 3 will pass rather than shrink. Since the new allocation of $\{4\}$ does not have a strictly larger sum of bids than the old one $\{1, 3\}$, it will be discarded, and bidder 4 will be asked again to either double its bid (to 16) or drop out. Since bidder 4's valuation is only 15, we expect it to drop out at this point. At this point every bidder is in either OLD $= \{1, 3\}$ or LOSERS $= \{2, 4\}$, so the shrinking auction terminates. Bidders 1 and 3 get the bundles $\{1, 2\}$ and $\{3, 4\}$, respectively, each for a price of 4. This allocation has welfare 18; the optimal welfare is 20.

## 4.4 Behavioral Assumptions

In Example 4.1, we frequently assumed that bidders behaved in the "obvious" way. We next formalize a set of behavioral assumptions that are sufficient to guarantee near-optimality of the allocation computed by the shrinking auction.

(B1) Every bidder $i$ drops out before its bid $b_i$ exceeds its value $v_i$. The justification for this assumption is that if bidder $i$ stays in the auction once $b_i > v_i$, it cannot receive positive utility — it either drops out later or receives a bundle at a price of at least $b_i > v_i$.

(B2) Conversely, no bidder $i$ will drop out until it is asked to double its bid to a value at least $v_i$. The justification for this assumption is that dropping out guarantees zero utility, while staying in the auction longer can only present opportunities to obtain nonnegative utility (as long as assumption (B1) is followed).

(B3) In line 4(a)ii of the shrinking auction, suppose (i) $b_i > v_i/2$; (ii) $S_i$ is infeasible (either $|S_i| > d$ or $S_i$ intersects with some $S_j$ with $j \in \textsc{new}$); and (iii) there is a set $T_i \subset S_i$ such that $i$ has value $v_i$ for $T_i$ and $T_i$ is feasible. Then bidder $i$ will shrink its set to some such $T_i$ (i.e., will not pass).[2] To justify this assumption, suppose (i)–(iii) hold. If $i$ passes, then at the end of the current iteration $i$ will be asked to double its bid. Since (i) holds and we are assuming (B1), $i$ will drop out and receive zero utility. If $i$ shrinks, on the other hand, it will never receive negative utility (assuming (B1)) and might receive positive utility (by (iii)).

(B4) No bidder $i$ will ever shrink its set $S_i$ to a set of items for which it has no value. Such an action would result in $i$ either dropping out later in the auction or receiving a worthless bundle for a positive price, leading to negative utility.

We trust that (B1)–(B4) are evidently reasonable assumptions. This intuition can also be made precise: for every strategy $\sigma$ of bidder $i$ that fails one of these assumptions, there is another strategy $\sigma'$ that dominates $\sigma$ and that does satisfy (B1)–(B4).[3] In other words, all undominated strategies satisfy (B1)–(B4). Moreover, every strategy $\sigma$ that fails one of (B1)–(B4) is "obviously dominated" — it is easy to compute a dominating strategy $\sigma'$ from $\sigma$.[4]

[2] We are implicitly assuming that each bidder $i$ can compute such a set $T_i$. This can be done in polynomial time if, for example, each bidder is only interested in a polynomial number of different bundles $A_{i\ell}$.

[3] That is, no matter what strategies the other bidders use, $i$'s utility under $\sigma'$ is at least that under $\sigma$. Also, there exist strategies for the other bidders such that $i$'s utility is strictly higher under $\sigma'$ then under $\sigma$.

[4] Babaioff et al. [1] highlight this property and for this reason call the shrinking auction an "algorithmic implementation in undominated strategies."

# 5 Analysis of Shrinking Auction

We now analyze the welfare achieved by the shrinking auction under our behavioral assumptions (B1)–(B4).[5] Recall from Section 4.2 that bidders have no dominant strategies, and a wide range of auction outcomes are possible under these assumptions. The following guarantees apply to every outcome that can arise under these assumptions.

Recall our assumption that all $v_i$'s are at least some publicly known positive value, which we're taking to be 1 for simplicity. The following welfare guarantee also depends on $v_{\max} = \max_{i=1}^{n} v_i$; $v_{\max}$ does not need to be publicly known.

**Theorem 5.1** *Under assumptions (B1)–(B4), the welfare of the allocation computed by the shrinking auction is at least an* $\Omega(1/(d \log^2 v_{\max}))$ *fraction of the maximum possible.*

Should you be impressed by Theorem 5.1? It depends. Without incentive constraints, the greedy algorithm obtains a $1/d$-approximation, and we can't expect to do much better with any polynomial-time algorithm or mechanism (Section 3). Theorem 5.1, or at least our analysis of it, suffers a loss of a further $\Theta(\log^2 v_{\max})$ factor. This is undesirable, and it would be nice to have a mechanism with an approximation factor that depends on $d$ only. On the other hand, we don't know how to get an approximation guarantee anywhere close to that in Theorem 5.1 with a DSIC mechanism. So relaxing the DSIC constraint to an implementation in undominated strategies seems to give a big win for Scenario #9, at least with respect to the current state-of-the-art.

We begin with a key lemma that bounds the number of iterations of the shrinking auction.

**Lemma 5.2** *Under assumption (B1), the shrinking auction terminates after at most* $3(\lceil \log_2 v_{\max} \rceil + 1)$ *iterations of the main while loop.*

*Proof:* At the end of every iteration except the last, at least one bidder either drops out or doubles its bid. Thus, under assumption (B1), the shrinking auction terminates in a finite number of iterations. Let $i$ denote the last bidder to drop out.[6]

If the bidder $i$ drops out with $S_i = U$, then it never shrank its set and it was never added to the set NEW. It was therefore forced to double its bid every iteration, and so by (B1) it dropped out after at most $\lceil \log_2 v_{\max} \rceil + 1$ iterations. For the rest of the proof, suppose that bidder $i$ drops outs with $S_i \neq U$. This implies that $i$ shrank its set in some iteration.

Consider first the case where $S_i$ intersects $S_j$ for some winning bidder $j$. Since bidders' sets only shrink throughout the shrinking auction, the sets of bidders $i$ and $j$ conflicted in every iteration of the auction. In each iteration, at least one of the two bidders was not allocated and hence asked to double its bid. Under assumption (B1), this implies an upper bound of $2(\lceil \log v_{\max} \rceil + 1)$ on the number of iterations.

[5] In Example 4.1, we made some additional assumptions, such as bidders 1 and 3 shrinking rather than passing in the first iteration. The analysis in Section 5 is valid assuming only (B1)–(B4).

[6] In the edge case where every bidder is allocated at termination, we can apply the following argument to a bidder $i$ that was not allocated in the penultimate iteration of the auction.

Finally, suppose that $S_i \neq U$ and $S_i$ is disjoint from $S_j$ from every winning bidder $j$. How could this have happened? This issue lies in step 4b. Before this line, $i$ was either in NEW or conflicted with the set $S_j$ of some other bidder $j$ in NEW. Since $i$ is neither allocated nor does it conflict with anyone who was allocated, it must be the case that OLD was chosen over NEW. Now rewind the shrinking auction to the iteration $t$ at which the allocation OLD was originally computed. Because $i$ is not in this allocation, its set $S'_i$ at iteration $t$ must have conflicted with the set $S_j$ of some bidder $j$ in the allocation. (By assumption, bidder $i$ has since shrunk its set from $S'_i$ to $S_i$ and no longer conflicts with any bidders in this allocation.) Since sets only shrink, bidders $i$ and $j$ were in conflict in every iteration up to iteration $t$. Thus, by the argument of the previous paragraph, $t \leq 2(\lceil \log v_{\max} \rceil + 1)$. Since the allocation OLD was chosen in iteration $t$ and in every subsequent iteration of the auction, and $i \notin$ OLD, $i$ was asked to double its bid in iteration $t$ and every iteration thereafter. Assumption (B1) then implies that there can be only $\lceil \log v_{\max} \rceil$ iterations after $t$, yielding a total iteration bound of $3(\lceil \log v_{\max} \rceil + 1)$. ■

Two more lemmas will yield Theorem 5.1. The high-level idea is to tease apart two different sources of welfare loss in the shrinking auction. For a fixed valuation profile, let $(S_1^*, \ldots, S_n^*)$ denote an optimal allocation. The first problem in the shrinking auction is that the shrunken sets $(\hat{S}_1, \ldots, \hat{S}_n)$ at termination need not contain $(S_1^*, \ldots, S_n^*)$. Thus, even the best-case allocation with $i$ receiving only items in $\hat{S}_i$ can have sub-optimal welfare. The second problem is that the shrinking auction only uses a greedy heuristic to compute the final allocation from the bundles $(\hat{S}_1, \ldots, \hat{S}_n)$. The next two lemmas quantify these two welfare losses.

**Lemma 5.3** *Let $(S_1^*, \ldots, S_n^*)$ denote an optimal allocation. Let the shrinking auction terminate with sets $(\hat{S}_1, \ldots, \hat{S}_n)$ after $R$ iterations. Let $(\hat{T}_1, \ldots, \hat{T}_n)$ denote an allocation that maximizes welfare under the constraint that each bidder $i$ only receives items from its set $\hat{S}_i$. Then, under assumption (B4),*

$$\sum_{i=1}^{n} v_i(\hat{T}_i) \geq \frac{1}{R+1} \cdot \sum_{i=1}^{n} v_i(S_i^*).$$

**Lemma 5.4** *Let the shrinking auction terminate with sets $(\hat{S}_1, \ldots, \hat{S}_n)$ after $R$ iterations, with winning bidders $W$. Let $(\hat{T}_1, \ldots, \hat{T}_n)$ denote an allocation that maximizes welfare under the constraint that each bidder $i$ only receives items from its set $\hat{S}_i$. Then, under assumptions (B2) and (B3),*

$$\sum_{i \in W} v_i(\hat{S}_i) \geq \frac{1}{2dR} \cdot \sum_{i=1}^{n} v_i(\hat{T}_i). \tag{1}$$

Observe that chaining together Lemmas 5.2–5.4 proves Theorem 5.1. We now provide the remaining two proofs.

*Proof of Lemma 5.3:* Let $F_r$ denote the bidders that shrink their set for the first time (from $U$ to a set $S_i^r$ of at most $d$ items) in iteration $r$. Let $F_0$ denote the bidders that never shrink their sets (and hence drop out with $S_i = U$). Note that $F_0, F_1, \ldots, F_R$ form a partition of the bidders.

A key observation is that, for each $r = 1, 2, \ldots, R$, the sets $\{S_i^r\}_{i \in F_r}$ are disjoint. The reason is that when a bidder shrinks its set, it must do so to a set that is feasible with respect to bidders that have already been added to NEW, and all bidders that previously shrunk their sets in an iteration already belong to NEW. Since sets only shrink over time, $\{\hat{S}_i\}_{i \in F_r}$ is a collection of disjoint sets for each $r \geq 1$. By assumption (B4), $v_i(\hat{S}_i) = v_i$ for each $i \in F_r$. Hence,

$$\sum_{i=1}^{n} v_i(\hat{T}_i) \geq \sum_{i \in F_r} v_i(\hat{S}_i) = \sum_{i \in F_r} v_i \tag{2}$$

for every $r = 1, 2, 3, \ldots, R$. Since the bundles in the optimal solution $(S_1^*, \ldots, S_n^*)$ are disjoint and $\hat{S}_i = U$ for all $i \in F_0$, we also have

$$\sum_{i=1}^{n} v_i(\hat{T}_i) \geq \sum_{i \in F_0} v_i(S_i^*). \tag{3}$$

Adding the $R$ inequalities (2) and the inequality (3) yields

$$(R+1) \sum_{i=1}^{n} v_i(\hat{T}_i) \geq \sum_{i \in F_0} v_i(S_i^*) + \sum_{i \notin F_0} v_i \geq \sum_{i=1}^{n} v_i(S_i^*),$$

and this implies the lemma. ■

*Proof of Lemma 5.4:* We can assume without loss of generality that $|\hat{T}_i| \leq d$ for every bidder $i$. Consider an iteration $r$ of the shrinking auction. The set $\text{NEW}^r$ is computed in this iteration using the current sets $S_1^r, \ldots, S_n^r$. The key claim is that

$$\sum_{i \in \text{NEW}^r} b_i^r \geq \frac{1}{d} \sum_{i \in L_r} b_i^r, \tag{4}$$

where $L_r$ is the set of bidders $i$ that drop out in iteration $r$ and that have $v_i(\hat{T}_i) > 0$. Intuitively, inequality (4) follows from the fact that the greedy algorithm is a $\frac{1}{d}$-approximation. There are some details in making this precise. As defined, the shrinking auction runs the greedy algorithm with respect to the $S_i^r$'s. More is true.

Assumption (B3) implies that when the shrinking auction reached bidder $i \in L_r$ in iteration $r$, all bundles that $i$ desires (including $\hat{T}_i$) were blocked by bidders previously added to $\text{NEW}^r$. Imagine we run the greedy algorithm on the bidders of $\text{NEW}^r \cup L_r$ with sets $S_i^r$ for $i \in \text{NEW}^r$ and sets $\hat{T}_i$ for $i \in L_r$. Note that all of these sets have size at most $d$. On this instance, the greedy algorithm will terminate with the allocation $\text{NEW}^r$. Since $(\hat{T}_1, \ldots, \hat{T}_n)$ is a feasible allocation, so is $\{\hat{T}_i\}_{i \in L_r}$. Since the greedy algorithm is a $\frac{1}{d}$-approximation, inequality (4) follows.

Since every bidder $i \in L_r$ dropped out in iteration $r$, assumption (B2) implies that $b_i^r \geq v_i/2$ for every $i \in L_r$ in iteration $r$ and hence (4) gives

$$\sum_{i \in \text{NEW}^r} b_i^r \geq \frac{1}{2d} \sum_{i \in L_r} v_i.$$

The monotonicity of the shrinking auction enforced in line 4b and assumption (B1) imply that

$$\sum_{i=1}^{n} v_i(\hat{S}_i) \geq \frac{1}{2d} \sum_{i \in L_r} v_i. \tag{5}$$

Letting $L_0$ denote the bidders that obtain a positive value in both $(\hat{T}_1, \ldots, \hat{T}_n)$ and $(\hat{S}_1, \ldots, \hat{S}_n)$, we obviously have

$$\sum_{i=1}^{n} v_i(\hat{S}_i) \geq \sum_{i \in L_0} v_i. \tag{6}$$

Since $L_0, \ldots, L_R$ are disjoint and $\cup_{r=0}^{R} L_r$ contains all bidders that obtain positive value in $(\hat{T}_1, \ldots, \hat{T}_n)$, we can sum (5) over all $R$ iterations and inequality (6) to obtain (1), completing the proof. ■

# References


[1] Moshe Babaioff, Ron Lavi, and Elan Pavlov. Single-value combinatorial auctions and algorithmic implementation in undominated strategies. *Journal of the ACM*, 56(1), 2009.

[2] Elad Hazan, Shmuel Safra, and Oded Schwartz. On the complexity of approximating $k$-set packing. *Computational Complexity*, 15(1):20–39, 2006.

[3] 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.