CS364B: Frontiers in Mechanism Design
Bonus Lecture: Gross Substitutes and Greedy Algorithms*

Tim Roughgarden†

February 7, 2014

1 Quick Review

We first recall the definition of a valuation that satisfies gross substitutes (GS). Recall that the *demand set* $D(\mathbf{p})$ of a valuation v given $\mathbf{p}$ on the items is $\text{argmax}_{S \subseteq U}\{v_i(S) - \sum_{j \in S} p_j\}$.

Definition 1.1 (Gross Substitutes) A valuation v_i defined on item set U satisfies the *gross substitutes (GS)* condition if and only if the following condition holds. For every price vector $\mathbf{p}$, every set $S \in D_i(\mathbf{p})$, and every price vector $\mathbf{q} \geq \mathbf{p}$, there is a set $T \subseteq U$ with

$$(S \setminus A) \cup T \in D_i(\mathbf{q}),$$

where $A = \{j : q(j) > p(j)\}$ is the of items whose prices have increased (in $\mathbf{q}$ relative to $\mathbf{p}$).

That is, whenever a bidder loses the items of $S \cap A$ because of being outbid, it still wants to retain the items $S \setminus A$ it has, at the original prices. We saw several examples of GS valuations, including k-unit demand valuations, downward-sloping valuations for identical items, and so on.

In this lecture, we insist that Definition 1.1 holds for all real-valued prices vectors, including those that have some negative prices. We also won't need to assume that $v_i(\emptyset) = 0$. We consider only valuations that are monotone ($S \subseteq T$ implies $v(S) \leq v(T)$).

 Thanks to Michal Feldman for several corrections to an earlier draft of these notes.

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

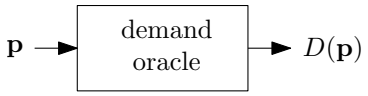

Figure 1: A demand oracle takes prices $\mathbf{p}$ as input and outputs a utility-maximizing set at these prices $\mathbf{p}$.

2 Greedy Solvability

The goal of this lecture is to prove that a valuation v satisfies the gross substitutes condition in Definition 1.1 if and only if it is "greedy solvable."[1] Recall that a *demand oracle* for a valuation v takes as input item prices $\mathbf{p}$ and outputs a member of the demand set $D(\mathbf{p})$ (Figure 1). Next is a greedy algorithm that attempts to implement a demand oracle. We use the notation

$$v(j|S) = v(S \cup j) - v(S)$$

for "j given S," the marginal value item j contributes to the set S. Later in the lecture, we use the notation $V(T|S) = V(T \cup S) - v(T)$ for the marginal value of a set T of items.

1. $S = \emptyset$
2. while (TRUE)
   (a) let $j \in \text{argmax}_U\{v(j|S) - p(j)\}$
   (b) if $v(j|S) \leq p(j)$, halt and return S
   (c) else add j to S

Definition 2.1 (Greedy Solvable) A valuation v_i defined on item set U is *greedy solvable* if, for every real-valued price vector $\mathbf{p}$, the greedy algorithm outputs a member of the demand set $D(\mathbf{p})$.

Analogous to Definition 1.1, we insist that the greedy algorithm is correct even for price vectors that contain negative prices.

The following result holds; see [6] for full details.

Theorem 2.2 *A valuation satisfies the gross substitutes condition if and only if it is greedy solvable.*

In this lecture, we give a full proof of the "only if" direction. We view this as the more immediately useful direction, though the converse is also very important. It reduces the understanding of gross substitutes valuations, which should seem mysterious and abstract in light of Definition 1.1, to understanding the highly non-mysterious and non-abstract concept

[1] Readers familiar with matroids and their characterization via optimality of the greedy algorithm with observe very strong parallels in today's lecture.

of greedy algorithms. Proving the "if" direction involves proving the converse of each of the steps of this lecture. The argument has a similar flavor and length.[2]

3 Examples and Applications

For a contrived but still helpful example, consider a weighted graph $(N, U, \mathbf{w})$ and view the edges U as items. Define a valuation v as follows: for $S \subseteq U$, define $v(S)$ as the maximum weight of an acyclic subgraph in the graph (N, S).[3] Given prices $\mathbf{p}$ (on edges), the greedy algorithm for this valuation boils down to Kruskal's minimum-spanning forest algorithm, with edge weights $\mathbf{w} - \mathbf{p}$. Correctness of Kruskal's algorithm implies that this valuation is greedy solvable, and hence (by Theorem 2.2) satisfies the gross substitutes condition.

A simple non-example is a single-minded valuation, say $v(S) = 1$ for $S = U$ and 0 otherwise, where $U = \{a, b\}$. We've seen previously that this valuation is not GS. It is also not greedy solvable: with the price vector $\mathbf{p} = (\epsilon, \epsilon)$, the greedy algorithm outputs $\emptyset$ while $D(\mathbf{p}) = \{U\}$.

Not many people have been thinking about gross substitutes valuations through the lens of greedy algorithms. That will change, however. Here are some immediate and potential applications of Theorem 2.2.

1. There are classes of valuations for which value queries (given S, what is $v(S)$?) can be answered in polynomial time, while answering a demand query (given $\mathbf{p}$, what is a bundle of $D(\mathbf{p})$?) is NP-hard.[4] Since the greedy algorithm in Section 2 can be implemented with a polynomial number of value queries, Theorem 2.2 implies that every GS valuation that supports polynomial-time value queries also supports polynomial-time demand queries.

2. In Lecture #6, we showed how to compute a welfare-maximizing allocation for bidders with gross substitutes valuations in polynomial time, provided the valuations support demand queries in polynomial time.[5] By the previous point, we can maximize welfare with GS valuations assuming only value queries. This answers an open question in [4]; see also [1].

3. The special structure of GS valuations allows the polynomial-time computation of a welfare-maximization allocation without recourse to linear programming; see [6, §10]. The algorithm by Murota [5] described in [6] is still quite slow, however — can our understanding of gross substitutes through greedy algorithms lead to a simpler or faster algorithm?

[2] Open question: can you find a simple and direct proof that greedy solvability implies the GS condition? Or at least that is implies submodularity?

[3] This is a special case of the "rank function of a weight matroid," a class of GS valuations.

[4] E.g., see the coverage valuations of Lecture #10.

[5] When we applied the ellipsoid algorithm to the dual linear program, the separation oracle was a demand query.

4. Recall from Lecture #5 that GS valuations are the limit for guaranteed existence of Walrasian equilibria: for every non-GS valuation v_i, there are unit-demand (and hence GS) valuations $\mathbf{v}_{-i}$ such that the valuation profile $\mathbf{v}$ admits no Walrasian equilibrium. Is there a simple proof of this using the "if" direction of Theorem 2.2? That is, given a valuation function for which the greedy algorithm can fail, can this failure directly be translated to a valuation profile without a Walrasian equilibrium?

4 High-Level Proof Steps

This section describes the high-level plan for proving the "only if" direction of Theorem 2.2. We first recall a result from Lecture #7 that will be useful in the proof: every GS valuation is submodular.

Proposition 4.1 ([3]) *If a valuation v satisfies the gross substitutes condition, then v is submodular.*

We proved this directly by invoking the GS condition for a series of price vectors to deduce the submodularity condition:

$$v(T \cup \{j\}) - v(T) \leq v(S \cup \{j\}) - v(S) \tag{1}$$

for every $j \in U$ and $S \subseteq T$.

For a valuation v and prices $\mathbf{p}$ on items U, the *greedy ordering* of U is the order in which the greedy algorithm chooses items, assuming that we force it to run for a full $|U|$ iterations, even after the marginal values of all remaining items have dropped below their prices (i.e., choosing each item exactly once). Ties can be broken arbitrarily. The main theorem that we prove is the following.[6]

Theorem 4.2 *Let v satisfy gross substitutes and let $\mathbf{p}$ be a real-valued price vector. Let $U = \{1, 2, \ldots, m\}$ denote the greedy ordering of the items. Then, for $t = 1, 2, \ldots, m$, the set S_t of the first t items maximizes*

$$v(T) - \sum_{j \in T} p_j$$

over all t-item subsets T of U.

Theorem 4.2 easily implies the "only if" direction of Theorem 2.2.

Corollary 4.3 *If v satisfies the gross substitutes condition, then v is greedy solvable.*

Proof: Theorem 4.2 immediately implies that one of the "prefix sets" S_t is a member of $D(\mathbf{p})$; the greedy algorithm outputs some prefix set S^*. To argue that the greedy algorithm stops with the optimal prefix set, first note its stopping rule implies that every item it adds strictly increases $v(S) - \sum_{j \in S} p_j$. Thus, "rolling back" S^* to some earlier prefix set can only

[6] The converse also holds; see [6].

reduce utility. The greedy algorithm's stopping rule also ensures that $v(j|S^*) \leq p(j)$ for every item $j \notin S^*$. Since v is submodular (Proposition 4.1), $v(j|S) \leq p(j)$ for all supersets S of S^*. This means that adding more items to S^* is always a bad idea, and no bigger prefix set has strictly larger utility than S^*. We conclude that $S^* \in D(\mathbf{p})$. ■

We now turn our attention to Theorem 4.2. The key technical lemma is the following, a sufficient condition for the optimality of greedy in the sense of Theorem 4.2.

Lemma 4.4 *Let v satisfy the gross substitutes condition. Then for every $S \subseteq U$, every $T \subseteq U \setminus S$ with $|T| \geq 2$, and every $i \in T$,*

$$v(i|S) + v(T \setminus \{i\}|S) \leq \max_{j \in T, j \neq i} \{v(j|S) + v(T \setminus \{j\}|S)\}. \tag{2}$$

Lemma 4.4 is hard to interpret the first time you see it. Note that an alternative way to explain (2) is: among the $|T|$ terms of the form $v(i|S) + v(T \setminus \{i\}|S)$ for $i \in T$, there is no unique maximum. This statement is trivial when $|T| = 2$ but is already interesting when $|T| = 3$.

We conclude this section by showing that the property in Lemma 4.4 is a sufficient condition for the optimality of the greedy algorithm in the sense of Theorem 4.2. The next section builds intuition for and proves the lemma.

Proof of Theorem 4.2: Fix a GS valuation v and a vector $\mathbf{p}$ of real-valued item prices. Let u denote the utility of a bundle, so

$$u(S) = v(S) - \sum_{j \in S} p_j.$$

The function u also satisfies the gross substitutes condition, since the relevant condition on demand sets in Definition 1.1 for two price vectors $\mathbf{p}', \mathbf{q}'$ is inherited from that of v with the price vectors $\mathbf{p}' + \mathbf{p}, \mathbf{q}' + \mathbf{p}$.

Fix $t \in \{1, 2, \ldots, m\}$; we show that the first t items S_t chosen by the greedy algorithm maximizes $u(\cdot)$ over all t-element subsets of U. We prove, by induction on $k = 1, 2, \ldots, t$, that the first k elements S_k chosen by the greedy algorithm belong to some t-element subset of U that maximizes $u(\cdot)$. The base case of $k = 1$ follows immediately from the definition of the greedy algorithm.

Assume that $1 < k \leq t$. By the inductive hypothesis, S_{k-1} belongs to a utility-maximizing t-element subset S^*. Let x_k denote the element of $S_k \setminus S_{k-1}$. If S^* contains x_k, then we are done. So, assume that $x_k \notin S^*$ and write $S^* = S_{k-1} \cup T^*$ (Figure 2). Taking $S = S_{k-1}$, $T = \{x_k\} \cup T^*$ (so $|T| \geq 2$), and $i = x_k$, Lemma 4.4 produces an item $j \in T \setminus \{x_k\} = T^*$ with

$$u(x_k|S_{k-1}) + u(T^*|S_{k-1}) \leq u(j|S_{k-1}) + u(T^* \cup \{x_k\} \setminus \{j\}|S_{k-1}).$$

By the definition of the greedy algorithm, $u(x_k|S_{k-1}) \geq u(j|S_{k-1})$. Hence,

$$u(T^*|S_{k-1}) \leq u(T^* \cup \{x_k\} \setminus \{j\}|S_{k-1}). \tag{3}$$

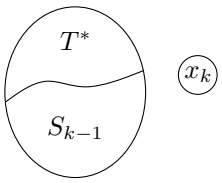

Figure 2: The set S^* is $S_{k-1} \cup T^*$ and $x_k \notin S^*$.

Adding $u(S_{k-1})$ to both sides of (3) yields $u(S^*) \leq u(S^* \cup \{x_k\} \setminus \{j\})$. Thus, $S^* \cup \{x_k\} \setminus \{j\}$ is also a utility-maximizing subset of t items, and it contains the first k items chosen by the greedy algorithm. This completes the inductive step and the proof of the theorem. ■

5 Proof of Lemma 4.4

We've reduced the result we really care about, Corollary 4.3, to the hard-to-interpret Lemma 4.4. The first interesting case of Lemma 4.4 is for sets T with $|T| = 3$, and we study this case in detail first. This is really the "essential" case of the lemma, in two senses. First, extending the proof for $|T| = 3$ to general T will be a slick but relatively painless induction. Second, the converse of Lemma 4.4 holds even if the condition 2 holds only for subsets T of size 3 (see [6]).

Writing $T = \{i, j, k\}$ for $T \subseteq U \setminus S$, Lemma 4.4 specializes to

$$v(i|S) + v(jk|S) \leq \max\left\{v(j|S) + v(ik|S), v(k|S) + v(ij|S)\right\}, \tag{4}$$

where for brevity we write $v(i|S)$ and $v(jk|S)$ for $v(\{i\}|S)$ and $v(\{j,k\}|S)$, respectively. That is, amongst the three ways to split up $\{i, j, k\}$ into a single and a pair and sum of the correspond marginal values (w.r.t. S), there is a tie for the maximum.

5.1 (Non-)Examples

For a non-example, consider the valuation

$$v(T) = \min\left\{3, \sum_{\ell \in T} v_\ell\right\}$$

defined on the item set $U = \{i, j, k\}$ with $v_i = 1$, $v_j = 2$, and $v_k = 3$. We showed in a different lecture (Lecture #7) that this valuation is submodular but does not satisfy the gross substitutes condition. We have

$$\begin{aligned} v(i|\emptyset) + v(jk|\emptyset) &= 4; \\ v(j|\emptyset) + v(ik|\emptyset) &= 5; \\ v(k|\emptyset) + v(ij|\emptyset) &= 6, \end{aligned}$$

which is a violation of the condition (4).

For an example, suppose v is a unit-demand valuation on the item set $U = \{i, j, k\}$:

$$v(T) = \max_{\ell \in T} v_\ell,$$

with $v_i = 1$, $v_j = 2$, and $v_k = 3$. Then

$$\begin{aligned} v(i|\emptyset) + v(jk|\emptyset) &= 4; \\ v(j|\emptyset) + v(ik|\emptyset) &= 5; \\ v(k|\emptyset) + v(ij|\emptyset) &= 5, \end{aligned}$$

and there is no obvious violation of (4). In general, with a unit-demand valuation, one of the three terms in (4) will be the minimum plus the maximum singleton valuation, while the other two will equal the median plus the maximum singleton valuations.

5.2 Proof of the $|T| = 3$ Case

We now prove the special case of Lemma 4.4 for sets $|T|$ of size 3. This is where we finally use the awkward hypothesis that a valuation v satisfies Definition 1.1, a task we've been eager to defer until now.

We prove the contrapositive. Suppose a valuation v violated (4), in the form of a set S and items $i, j, k \notin S$ with

$$v(i|S) + v(jk|S) > v(j|S) + v(ik|S) \tag{5}$$

and

$$v(i|S) + v(jk|S) > v(k|S) + v(ij|S). \tag{6}$$

If v is not submodular, then it is not gross substitutes (Proposition 4.1) and we are done. Henceforth, we assume that v is submodular.

Claim: There is a price vector $\mathbf{p}$ such that $D(\mathbf{p}) = \{S \cup \{i\}, S \cup \{j, k\}\}$.

The claim implies that v does not satisfy the gross substitutes condition and hence proves the lemma (in the special case where $|T| = 3$). To see this, observe that at the prices $\mathbf{p}$, $S \cup \{i\}$ is the unique utility-maximizing bundle among those excluding j, $S \cup \{j, k\}$ is the unique utility-maximizing bundle among those including j, and these two bundles have equal utility. After increasing the price of item j, $S \cup \{i\}$ is the unique utility-maximizing bundle. A bidder with preferred bundle $S \cup \{j, k\}$ would want to relinquish item k after the price of item j is increased, a violation of Definition 1.1.

Proof of Claim: We define the price vector $\mathbf{p}$ as follows. Since we want all items of S and no items outside $S \cup \{i, j, k\}$ to be demanded, we set $p(\ell) = -\infty$ (or sufficiently negative) for $\ell \in S$ and $p(\ell) = +\infty$ (or sufficiently positive) for $\ell \notin S \cup \{i, j, k\}$.

Since we want $S \cup \{j, k\}$ to give strictly higher utility than either $S \cup \{j\}$ or $S \cup \{k\}$, we define $p(j) = v(j|S \cup \{k\}) - \epsilon$ and $p(k) = v(j|S \cup \{j\}) - \epsilon$, where ϵ is a sufficiently small

positive number. Since v is submodular, these prices also ensure that S yields strictly less utility than $S \cup \{j,k\}$.

Next, we define $p(i)$ so that the bundle $S \cup \{i\}$ gives the same utility as $S \cup \{j,k\}$. To finish the proof, we argue that the utility of $S \cup \{i\}$ is strictly better than that of $S \cup \{i,j\}$ or $S \cup \{i,k\}$; submodularity of v then implies that it is also yields strictly higher utility than $S \cup \{i,j,k\}$, which shows that $D(\mathbf{p}) = \{S \cup \{i\}, S \cup \{j,k\}\}$, as desired.

The difference between the utility of $S \cup \{i,j\}$ and the utility of $S \cup \{i\}$ is

$$\begin{aligned} v(j|S \cup \{i\}) - p(j) &= v(j|S \cup \{i\}) - v(j|S \cup \{k\}) + \epsilon \\ &= v(ij|S) - v(i|S) - v(jk|S) + v(k|S) + \epsilon \\ &< 0, \end{aligned}$$

where the inequality follows from (6), provided ϵ is sufficiently small. Similarly, the inequality (5) implies that the bundle $S \cup \{i,k\}$ yields strictly less utility than $S \cup \{i\}$. This completes the proof of the claim and the $|T| = 3$ case of Lemma 4.4.

5.3 A Useful Corollary

We state an immediate corollary of the $|T| = 3$ case of Lemma 4.4 that is useful in the inductive argument in the next section and also interesting in its own right.[7] Multiplying the inequality in (4) by -1 and adding $v(i|S) + v(j|S) + v(k|S)$ to both sides yields the following.

Corollary 5.1 *If v satisfies the gross substitutes conditions, then for every subset S of items and items $i,j,k \notin S$,*

$$\underbrace{v(j|S) + v(k|S) - v(jk|S)}_{\alpha_S(j,k)} \geq \min \left\{ \underbrace{v(i|S) + v(k|S) - v(ik|S)}_{\alpha_S(i,k)}, \underbrace{v(j|S) + v(k|S) - v(jk|S)}_{\alpha_S(j,k)} . \right\} \tag{7}$$

We use the notation $\alpha_S(i,j)$ for $v(i|S) + v(j|S) - v(ij|S)$, which can be interpreted as "measure of substitutability" for the items i and j (given the possession of items S). When v is monotone and submodular, $\alpha_S(i,j)$ can be as small as 0 (when v acts additively on i and j given S) and as large as $\min\{v(i|S), v(j|S)\}$ (when v acts like a unit-demand valuation on i and j given S). Corollary 5.1 states that for gross substitutes valuations, among every triple i,j,k of items outside a set S, there must be a tie for the minimum α_S-value.

5.4 Completing the Proof of Lemma 4.4

We need to show that for all $S \subseteq U$, $T \subseteq U \setminus S$ with $|T| \geq 2$, and $i \in T$, there is a $j \in T \setminus \{i\}$ with

$$v(i|S) + v(T \setminus \{i\}|S) \leq v(j|S) + v(T \setminus \{j\}|S). \tag{8}$$

[7] It is also the key property that permits extension of the positive results in Lecture #10 for coverage valuations to the convex hull of gross substitutes valuations [2].

We proceed by induction on T. The case $|T| = 2$ is trivial and the case $|T| = 3$ was proved in Section 5.2. For the inductive step, fix $t \geq 4$ and assume that (8) holds whenever $|T| \leq t - 1$.

Fix S, $T \subseteq U \setminus S$ with $|T| = t$, and $i \in T$. An obvious idea is to delete an element k from $T \setminus \{i\}$ and apply the inductive hypothesis. Two non-obvious ideas are to also add k to S, to minimize distance between the inequality we want and the one we get from the inductive hypothesis (see (9) below), and to choose k carefully (detailed below). For now, we pick $k \in T \setminus \{i\}$ arbitrarily and apply the inductive hypothesis to $S \cup \{k\}$, $T \setminus \{k\}$, and i to produce an item $j \in T \setminus \{i, k\}$ that satisfies

$$v(i|S \cup \{k\}) + v(T \setminus \{i, k\}|S \cup \{k\}) \leq v(j|S \cup \{k\}) + v(T \setminus \{j, k\}|S \cup \{k\}). \tag{9}$$

Adding $2v(k|S)$ to both sides of (9) yields an inequality closer to the one (8) that we want:

$$v(ik|S) + v(T \setminus \{i\}|S) \leq v(jk|S) + v(T \setminus \{j\}|S). \tag{10}$$

Comparing (8) with (10), if we only had

$$v(i|S) - v(ik|S) \leq v(j|S) - v(jk|S) \tag{11}$$

or, in the notation of Corollary 5.1,

$$\alpha_S(i, k) \leq \alpha_S(j, k), \tag{12}$$

then we'd be done (since adding (10) and (11) yields (8)).

To complete the proof of the inductive step and the lemma, we claim that if we choose k minimizing $\alpha_S(i, \ell)$ over $\ell \in T \setminus \{i\}$, then (12) holds. To see this, consider $\alpha_S(i, j)$, $\alpha_S(i, k)$, and $\alpha_S(j, k)$. By the choice of k, $\alpha_S(i, k) \leq \alpha_S(i, j)$. Since there is no unique minimum among the three values (Corollary 5.1), $\alpha_S(i, k) \leq \alpha_S(j, k)$ as well, verifying (12) and completing the proof.

References

[1] A. Bertlensen. Substitutes valuations and $M^\natural$-concavity. Hebrew University of Jerusalem, M.S. thesis, 2004.

[2] S. Dughmi, T. Roughgarden, and Q. Yan. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 149–158, 2011.

[3] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.

[4] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[5] K. Murota. Valuated matroid intersection II: Algorithms. *SIAM Journal on Discrete Mathematics*, 9(4):562–576, 1996.

[6] R. Paes Leme. Gross substitutability: an algorithmic survey. Working paper, 2013.