GROSS SUBSTITUTABILITY : AN ALGORITHMIC SURVEY

RENATO PAES LEME*

Abstract. The concept of gross substitute valuations was introduced by Kelso and Crawford as a sufficient conditions for the existence of Walrasian equilibria in economies with indivisible goods. The proof is algorithmic in nature: gross substitutes is exactly the condition that enables a natural price adjustment procedure – known as *Walrasian tatônnement* – to converge to equilibrium.

The same concept was also introduced independently in other communities with different names: $M^\natural$-concave functions (Murota and Shioura), Matroidal and Well-Layered maps (Dress and Terhalle) and valuated matroids (Dress and Wenzel). Here we survey various definitions of gross subtitutability and show their equivalence. We focus on algorithmic aspects of the various definitions. In particular, we highlight that gross substitutes are the exact class of valuations for which demand oracles can be computed via an ascending greedy algorithm. It also corresponds to a natural discrete analogue of concave functions: local maximizers correspond to global maximizers.

Finally, we discuss algorithms for the welfare problem (computing an optimal allocation of a set of items when agents have gross substitute valuations) as well as the related problem of computing Walrasian prices. We discuss approximation schemes based on the tatônnement procedure, linear programming approaches and purely combinatorial strongly-polynomial time algorithms.

1. Gross substitutes and Walrasian tatônnement.

The notion of *gross substitutes* was introduced by Kelso and Crawford [26] in order to analyze two sided matching markets of workers and firms. Originally it was defined as a condition on the *gross product* generated by a set of workers for a given firm, hence the name *gross substitutes*. Such condition allowed a natural salary adjustment process to converge to a point where each worker is hired by some firm and no worker is over-demanded. Gul and Stacchetti [21] later use the same notion to analyze the existence of price equilibria in markets with indivisible goods. For this survey, we adopt the Gul and Stacchetti terminology and talk about buyers/items/prices instead of firms/workers/salaries as in Kelso and Crawford.

Before we proceed, we fix some notation: we denote by $[n] = \{1, \ldots, n\}$ a set of items (goods). A *valuation* over such items is a function $v : 2^{[n]} \to \mathbb{R}$ such that $v(\emptyset) = 0$[1]. Given a price vector $p \in \mathbb{R}^n$ and a set $S \subseteq [n]$, we denote $p(S) = \sum_{j \in S} p_j$. We will define $v_p(S) = v(S) - p(S)$ as the value of a subset S under the price vector p. This corresponds to the utility of an agent with this valuation for acquiring such set under those prices. Given disjoint sets S, T we define the *marginal value* of T with respect to S as $v(T|S) = v(T \cup S) - v(S)$. We sometimes omit braces in the representation of sets when this is clear from the context, for example, by $v(i, j|S)$ we denote $v(\{i, j\}|S)$ and by $S \cup j$ we denote $S \cup \{j\}$.

An *economy with indivisible goods* is composed by a set $[n]$ of items (goods) and $[m]$ of buyers (agents) where each agent $i \in [m]$ has a valuation $v^i : 2^{[n]} \to \mathbb{R}$. We use the notion of the *demand correspondence* to define an equilibrium of this economy:

DEFINITION 1.1 (demand correspondence). *Given a valuation function $v : 2^{[n]} \to \mathbb{R}$ and a vector of prices $p \in \mathbb{R}^n$, we define the demand correspondence as the family of sets that maximize the utility of an agent under a price vector p :*

$$D(v, p) := \{S \subseteq [n]; v_p(S) \geq v_p(T), \forall T \subseteq [n]\}$$

*Google Research NYC, (renatoppl@google.com).

[1]Note that we don't require monotonicity in the definition. When we refer to a valuation for which $v(S) \leq v(T)$ whenever $S \subseteq T$, we will refer to it as a *monotone* valuations or valuations satisfying *free-disposal*.

DEFINITION 1.2 (Walrasian equilibrium). *Given an economy with indivisible goods with n goods, m agents and valuations $\{v^i\}_i$ satisfying free-disposal*[2]*, a* Walrasian equilibrium *corresponds to a vector of prices $p \in \mathbb{R}^n_+$ and a partition of the goods in disjoint sets $[n] = \cup_{i=1}^m S_i$ such that $S_i \in D(v^i, p)$ for all i.*

A reader familiar with the duality theorem in linear programming will readily recognize that the definition of Walrasian equilibrium closely resembles the *complementarity conditions* where the prices play the role of dual variables. Indeed, this is formalized by the results known as the First and Second Welfare Theorem. The First Welfare Theorem states that if $(p, S_1, \ldots, S_m)$ is a Walrasian equilibrium then this partition corresponds to the optimal allocation of goods, i.e., the allocation maximizing $\sum_i v^i(S_i)$. The proof is quite elementary: let $S_1^*, \ldots, S_m^*$ be any partition maximizing the welfare. Then since $S_i \in D(v^i, p)$, it must be the case that: $v^i(S_i) - p(S_i) \geq v^i(S_i^*) - p(S_i^*)$. Summing for all i and observing that $\sum_i p(S_i) = p([n]) = \sum_i p(S_i^*)$, we conclude that $\sum_i v^i(S_i) \geq \sum_i v^i(S_i^*)$.

The analogy with linear programming is completed by what is called the Second Welfare Theorem. It states that if $(p, S_1, \ldots, S_m)$ is a Walrasian equilibrium and $S_1^*, \ldots, S_m^*$ maximizes $\sum_i v^i(S_i^*)$, then $(p, S_1^*, \ldots, S_m^*)$ is also a Walrasian equilibrium. The proof is also simple, observe that summing $v^i(S_i) - p(S_i) \geq v^i(S_i^*) - p(S_i^*)$ for all i we obtain $\sum_i v^i(S_i) \geq \sum_i v^i(S_i^*)$. But since this is an equality, we should have an equality for each agent i: $v^i(S_i) - p(S_i) = v^i(S_i^*) - p(S_i^*)$, hence $S_i^* \in D(v^i, p)$.

A natural question is for which economies there exist Walrasian equilibria. Kelso and Crawford define a very natural price adjustment procedure and define gross substitutes as the natural sufficient condition for such process to converge. The general idea behind this procedure goes back to Walras' *tatônnement procedure* [51], where tatônnement means *trial-and-error*. The idea is that we start with an arbitrary price vector and compute one set in the demand of each agent. Then, for each item that is demanded by more then one agent (over-demanded) we increase the price. For each item that is demanded by no agent (under-demanded), we decrease the price. We iterate this until no item is over-demanded or under-demanded.

Let's describe this procedure precisely. We will make some modifications to the idea above to make the procedure simpler to analyze. Instead of starting from an arbitrary price vector p, we will start with zero prices for all items and only allow prices to increase. Moreover, we will start with all the items allocated to the first player at zero price and we will take turns asking buyers to choose their favorite set of items given prices as follows: the current price p_j for items currently allocated to him and $p_j + \delta$ for items allocated to other players. Once he takes items from other players, the prices of such items increase by δ.

Notice that the procedure has to stop at some point, since prices cannot increase indefinitely. If the price of an item is higher then $\max_{i,S} v^i(S)$, for example, no agent will demand this item and the price will freeze. Let p be the final price and p^i be the price faces by each agent. It should be the case that $S_i \in D(v^i, p^i)$, which means that for all $T \subseteq [n]$, $v^i(S_i) - p^i(S_i) \geq v^i(T) - p^i(T)$. This can be re-written as: $v^i(S_i) - p(S_i) \geq v^i(T) - p(T) - \delta|T \setminus S_i|$.

[2]The definition of Walrasian equilibrium can be changed to incorporate valuations not satisfying free-disposal. We do so by partitioning the items in $m+1$ disjoint sets $S_0, S_1, \ldots, S_m$. The items in S_0 are not allocated and are required to be priced at zero at in equilibrium.

ALGORITHM 1: Walrasian tatônnement procedure

Input: $\delta > 0$, $n, m \in \mathbb{Z}_+$ and v^i for $i \in [m]$
Set zero prices for all items: $p_j = 0, \forall j \in [n]$
Set initial allocation $S_1 = [n], S_i = \emptyset, \forall i \in [m] \setminus \{1\}$
Implicitly define $p^i \in \mathbb{R}^n$ as a function of p s.t. $p^i_j = p_j$ if $j \in S_i$ and $p^i_j = p_j + \delta$ o.w.
while there exists i such that $S_i \notin D(v^i, p^i)$
    find a demanded set under the p^i price vector $X_i \in D(v^i, p^i)$
    update prices: for $j \in X_i \setminus S_i$, set $p_j = p_j + \delta$ (vectors p^i are implicitly updated)
    update allocations: $S_i = X_i$ and $S_j = S_j \setminus X_i$ for $j \neq i$

In the limit as $\delta \to 0$, we recover a price vector and allocation such that $v^i(S_i) - p(S_i) \geq v^i(T) - p(T)$. To make the previous statement precise, let $(p^t, S^t_1, \ldots, S^t_m)$ be the outcome of the Walrasian tatônnement procedure for $\delta_t = \frac{1}{t}$ for $t \in \mathbb{Z}_+$. Since there are finitely many allocations $(S^t_1, \ldots, S^t_m)$, there is one allocation that happens infinitely often. Let $S_1, \ldots, S_m$ be such allocation and let $t_1 < t_2 < \ldots$ be the infinite subsequence corresponding to this allocation. Since p^t is bounded, passing to a subsequence if necessary, we can assume that $p^t \to p$. So taking $t \to \infty$ for this subsequence, we get $v^i(S_i) - p(S_i) \geq v^i(T) - p(T)$ for all i and $T \subseteq [n]$.

The argument above gives us an existential proof of a price vector p and an allocation S_i such that each agent is getting his optimal bundle under the current prices. This is not yet a Walrasian equilibrium, since Definition 1.2 requires the allocation to be a partition of the set of items, i.e., $\cup_i S_i = [n]$. The definition of gross substitutability is exactly what is needed to ensure that we can run the procedure above in such a way that all the items are allocated in the end. Since we started the Walrasian tatônnement procedure with a partition of the items, if we can always find a demanded set $X_i \in D(v^i, p^i)$ containing his currently allocated items, i.e., $S_i \subseteq X_i$, then we can guarantee the invariant that no item is even un-allocated during the execution of the algorithm. This motivates the following definition:

Definition 1.3 (gross substitutes, Kelso and Crawford [26]). *A valuation function satisfies the gross substitutes property if for any price vectors $p \in \mathbb{R}^n$ and $S \in D(v, p)$, if p' is a price vector with $p \leq p'$, then there is a set $S' \in D(v, p')$ such that $S \cap \{j; p_j = p'_j\} \subseteq S'$.*

In other words: if an agent with a gross substitute valuation demands a set S of items under a price vector p and the price of some items subsequently increase, the agent still has a demanded set that contains the items in S whose price didn't increase.

Theorem 1.4 (Kelso and Crawford [26]). *If valuations $v^1, \ldots, v^m$ satisfy the gross substitutes property, then a Walrasian equilibrium always exists.*

In some sense, gross substitutability is also necessary for the existence of Walrasian equilibria. Gul and Stacchetti [21] show the following: let $\mathcal{C}$ be a class of valuation functions that contains all *unit demand valuations*, i.e., all valuations of the type $v(S) = \max_{j \in S} v(\{j\})$. Then if $\mathcal{C}$ is such that for all $v^1, \ldots, v^m \in \mathcal{C}$ there is a Walrasian equilibrium, then all valuations in $\mathcal{C}$ are gross substitutes.

2. Examples, Non-Examples and Submodularity. It is instructive to have in mind a couple of examples and non-examples of gross substitute functions, to guide our intuition in the following sections. Three simple classes of valuations that can be readily recognized as gross substitutes from Definition 1.3 are:

1. *additive valuations*, i.e., valuations for which $v(S) = \sum_{i \in S} v(\{i\})$
2. *unit-demand valuations*, i.e., valuations for which $v(S) = \max_{i \in S} v(\{i\})$
3. *symmetric concave valuations*, i.e., valuations of the form $v(S) = f(|S|)$ for some monotone concave function $f : \mathbb{R}_+ \to \mathbb{R}_+$.

A less obvious example is the class of valuations known as *assignment valuations* introduced by Shapley [47], also called OXS valuations by Lehmann, Lehmann and Nisan [31]. An assignment valuation with n items can be represented as an $n \times k$ matrix V with non-negative entries, where each row corresponds to an item and each column to a position. Entry $V_{i,j}$ corresponds to the value of item i in position j. Each item can be assigned to a single position and each position can be assigned only one item. The value of a subset S of items is the value of the optimal assignment of items in S to positions. Items and positions are allowed to be left unassigned. Mathematically:

$$v(S) = \max\{\textstyle\sum_{i \in S} \sum_j V_{ij} \cdot x_{ij} \text{ s.t. } \sum_k x_{kj} \leq 1, \sum_k x_{ik} \leq 1 \text{ and } x_{ij} \in \{0,1\}, \forall i, j\}$$

We postpone until Section 9 a proof that all assignment valuations are gross substitutes, but we remark that unit-demand functions are the special case where there is only one position, additive functions correspond to the case where the matrix is an $n \times n$ diagonal matrix and the symmetric concave valuations to the case in which all the rows of the matrix are equal.

Hatfield and Milgrom [1] define a generalization of assignment valuations, which they call *endowed assignment valuations*. The show that this class also satisfied the gross substitutes property and that it captures most practical applications of gross substitutes valuations in matching markets. A valuation v on n items is said to be an endowed assignment valuation if there is an assignment valuation w on a larger set of $n + t$ items such that $v(S) = w(S|T)$ where S is a subset of the first n items and T corresponds to the last t items.

Another important example of gross substitutes is the class of *matroid rank functions*. This survey won't formally require any knowledge of matroid theory, but basic familiarity with matroids will be useful to guide the reader's intuition later on. The topic is too extensive to survey here, but we point to Lawler [30], Oxley [42] or Schrijver [46] for a comprehensive discussion.

We remark that even though matroid rank functions are gross substitute valuations, sums of matroid rank functions might *not* be. For example, given three items $\{a, b, c\}$ define for each $i \in \{a, b, c\}$, the function $r^i : 2^{\{a,b,c\}} \to \mathbb{R}$ such that: $r^i(\emptyset) = 0$, $r^i(S) = 1$ for $|S| = 1$, $r^i(\{a, b, c\} \setminus i) = 1$, and $r^i(S) = 2$ for all remaining subsets S. Notice that for each i, r^i is a matroid rank function and hence satisfy the gross substitutability. However, the valuation $v = r^a + 2 \cdot r^b + 3 \cdot r^c$ does not satisfy gross substitutability. In order to see that, observe that for the price vector $p = [4, 5, 4]$, $D(v; p) = \{\{a\}, \{c\}, \{a, c\}, \{b, c\}\}$. If the price of item c increases to ∞, i.e., $p' = [4, 5, \infty]$, then demand set becomes $D(v; p') = \{\{a\}\}$. So the increase in price of item c makes item b no longer belong to any demanded set, violating Definition 1.3.

Gul and Stachetti [21] observe that gross substitutes are a subclass of *submodular functions*:

DEFINITION 2.1 (submodularity). *A valuation function is said to be* submodular *if for all subsets* $S, T \subseteq [n]$, $v(S \cap T) + v(S \cup T) \leq v(S) + v(T)$. *Equivalently, for every* $S \subseteq [n]$ *and* $i, j \notin S$, $v(i, j|S) \leq v(i|S) + v(j|S)$.

THEOREM 2.2 (Gul and Stacchetti [21]). *Every gross substitute valuation function is submodular.*

Proof. Let v be a gross substitute valuation function. Given $S \subseteq [n]$ and $i, j \notin S$, consider the price vector[3] such that $p_t = \infty$ for $t \notin S \cup ij$, $p_t = -\infty$ for $t \in S \cup j$ and $p_i = v(i|S \cup j)$. Clearly $S \cup ij \in D(v, p)$. Now, if one defines p' such that $p'_j = \infty$ and $p'_t = p_t$ for all other t, then by gross substitutability, $S \cup i$ must be a demanded set. Therefore: $v(i|S) \geq v(i|S \cup j)$. Expanding this expression we get $v(S \cup i) - v(S) \geq v(S \cup ij) - v(S \cup j)$. Subtracting $2v(S)$ from each side and rearranging terms we obtain the definition of submodularity $v(i, j|S) \leq v(i|S) + v(j|S)$. □

Since the example $v = r^a + 2 \cdot r^b + 3 \cdot r^c$ earlier in this section is the sum of matroid rank functions, and hence submodular, it is clear that gross substitutes is a strict subclass of submodular functions. Another good source of examples of submodular but not gross substitute functions comes from the class of budget additive functions. We say that a valuation function is budget additive if it is of the form: $v(S) = \min\{B, \sum_{i \in S} w_i\}$ for non-negative real numbers $B, w_1, \ldots, w_n$. The following example due to Lehmann, Lehmann and Nisan [31] shows function that is budget additive but not gross substitutes: consider three items $\{a, b, c\}$ with weights $w_a = w_b = 1$ and $w_c = 2$ and budget $B = 2$. In order to see that the associated budget additive function is not gross substitutes, notice that for the prices $p = [\frac{1}{2}, \frac{1}{2}, 1]$, $D(v, p) = \{\{a, b\}, \{c\}\}$, but if the price of a increases and the price vector becomes $p' = [1, \frac{1}{2}, 1]$, then the demand correspondence becomes $D(v, p') = \{\{c\}\}$, i.e., the increase in the price of a makes item b be no longer demanded.

3. Gross substitutes, greedy demand oracles and local search. An important property of gross substitute valuations is that it is the exact class of functions for which demand oracles can be computed via a greedy or local search algorithms. In this section we make this statement precise by defining *matroidal maps* and *discrete concave valuations*. The first part of the survey will be devoted to show that those three concepts are equivalent.

An algorithmic primitive needed to implement the Walrasian tatônnement procedure is the computation of a set in the demand correspondence $X \in D(v, p)$. This is usually referred as the *demand oracle problem*. A simple heuristic to compute demand oracles is the *greedy algorithm*: start with the empty set X and keep adding the element $j \notin X$ that gives the maximum improvement to $v_p(X)$. In other words:

[3] allowing prices to take values ∞ and $-\infty$ simplifies the arguments. To be more precise, one can view such prices as M or $-M$ for $M = 1 + \max_S v(S)$.

ALGORITHM 2: Greedy demand oracle

Input: $p \in \mathbb{R}^n_+$, $v : 2^{[n]} \to \mathbb{R}_+$
Initialize $X = \emptyset$
repeat
find $j^* \in [n] \setminus X$ maximizing $\Delta_j = v(j|X) - p_j$
if $\Delta_{j^*} > 0$, $X = X \cup \{j^*\}$
if $X = [n]$ or $\Delta_{j^*} \leq 0$, **return** X

DEFINITION 3.1 (matroidal map). *A valuation function is a* matroidal map *if for all price vectors $p \in \mathbb{R}^n$, the greedy algorithm implements a demand oracle, i.e., $\mathrm{G}(v,p) \in D(v,p)$, where $\mathrm{G}(v,p)$ is the output of Algorithm 2.*

THEOREM 3.2. *A valuation function satisfies the gross substitute property if and only if it is a matroidal map.*

Before we proceed with the task of proving the previous theorem, we also mention another algorithmic definition of gross substitutability based on local search. Consider the following heuristic to compute demand oracles: start at an arbitrary set X and try to find a set improving v_p in the neighborhood $\mathcal{N}$ of X, where the neighborhood is composed by all sets that can be obtained from X by adding one element, removing one element or exchanging an element in the set by one element outside the set.

ALGORITHM 3: Local search demand oracle

Input: $p \in \mathbb{R}^n_+$, $v : 2^{[n]} \to \mathbb{R}_+$, $X_0 \subseteq [n]$
Initialize $X = X_0$
repeat
Let $\mathcal{N} = \{X \cup \{i\}; i \notin X\} \cup \{X \setminus \{i\}; i \in X\} \cup \{X \cup \{i\} \setminus \{i'\}; i \notin X, i' \in X\}$
If $\max_{Y \in \mathcal{N}} v_p(Y) \leq v_p(X)$, **return** X
Else choose some $Y \in \mathcal{N}$ with $v_p(Y) > v_p(X)$ and let $X = Y$.

Gul and Stacchetti [21] show that yet another way of defining gross substitutes is as the class of valuation functions for which local search is exact, i.e., it doesn't get stuck on local minima:

DEFINITION 3.3 (discrete concave valuation). *A valuation function is discrete concave if for all price vectors $p \in \mathbb{R}^n$, the local search algorithm implements a demand oracle, i.e., $\mathrm{L}(v,p,X_0) \in D(v,p)$, where $\mathrm{L}(v,p,X_0)$ is the output of Algorithm 3.*

Equivalently, a valuation function is discrete concave if and only if for all price vectors $p \in \mathbb{R}^n$, $S \in D(v,p)$ iff $v_p(S) \geq v_p(S \cup i)$, $v_p(S) \geq v_p(S \setminus j)$ and $v_p(S) \geq v_p(S \cup i \setminus j)$, for all $i \notin S$ and $j \in S$.

THEOREM 3.4 (Gul and Stacchetti [21]). *A valuation function satisfies the gross substitute property if and only if it is discrete concave.*

Roadmap. The first part of the survey will be devoted to prove the algorithmic characterizations of gross substitutes given by Theorems 3.2 and 3.4. We will follow a somewhat indirect route towards this goal outlined in Figure 6.1, through a price independent characterization (Section 4), the form of the Hessian (Section 5) and the concept of well-layered maps (Section 6). Once the appropriate tools are established,

it will be possible to give a simple proof of Theorems 3.2 and 3.4 as a consequence of more fundamental properties of those functions. In Section 8 we discuss connections to Discrete Convex Analysis. Some of the proofs in the first half of the survey, specially in Sections 4, 6 and 8 are somewhat technical and I suggest the first-time reader to familiarize himself with the structural results but skip the proofs. The second part of the survey (Sections 9, 10 and 11) deals with the computation of Walrasian equilibrium.

4. A price independent characterization. We make a brief detour and look at a different, yet very related question about gross substitutes. In the previous section we gave two alternative algorithmic characterizations of gross substitutes. All characterizations given so far involve prices, i.e., they are of the form: a valuation v satisfied the gross substitutes property if for all price vectors p, the pair (v, p) has some given property. The question of giving an explicit characterization of gross substitutes was resolved simultaneously by Fujishige and Yang [19] and Reijnierse, Gellekom and Potters [44]. The first paper provides a powerful connection to the theory of Discrete Convex Analysis, which we discuss in more detail in Section 8. We focus first on the definition given by Reijnierse et al [44].

THEOREM 4.1 (Reijnierse, Gellekom and Potters [44]). *A valuation function has the gross substitutes property iff it is submodular and for all sets $S \subseteq [n]$ and all distinct $i, j, k \notin S$, the following holds:*

$$v(i, j|S) + v(k|S) \leq \max[\ v(i|S) + v(j, k|S), v(j|S) + v(i, k|S)\ ] \qquad \text{(RGP)}$$

The first step involves showing that if a function is not gross substitutes there is a simple certificate given by the following lemma. The proof of the lemma is quite technical and not particularly enlightening and therefore omitted here.

LEMMA 4.2. *A valuation function v is in gross substitutes iff there is no price vector $p \in \mathbb{R}^n$ such that*

$$\textit{either (i) } D(v,p) = \{S, S \cup ij\} \textit{ or (ii) } D(v,p) = \{S \cup k, S \cup ij\}.$$

Proof of Theorem 4.1. We now transform the existence of certificates of Lemma 4.2 in simple conditions on v. This is based on two observations: There exists a certificate of type (i) iff the v is not submodular. There exists a certificate of type (ii) iff there is a violation of condition (RGP).

First, consider a certificate of type (i). So, there are prices such that $0 = v(i, j|S) - p_i - p_j > \max[v(i|S) - p_i, v(j|S) - p_j]$. Summing the inequalities $0 > v(i|S) - p_i$ and $v(i, j|S) - p_i - p_j > v(j|S) - p_j$ we get: $v(i, j|S) > v(i|S) - v(j|S)$ which is a violation of submodularity. Conversely, if you have a violation of submodularity for i, j, S, take $p_t = -\infty$ for $t \in S$, $p_t = \infty$ for $t \notin S \cup \{i, j\}$ and $p_i = v(i|S) + \epsilon$ and $p_j = v(j|S) + \epsilon$ for some tiny ϵ and this gives us a certificate of type (i).

Consider now a certificate of type (ii). So, there are prices such that $v(k|S) - p_k = v(i, j|S) - p_i - p_j > \max[v(i|S) - p_i, v(j|S) - p_j, v(i, k|S) - p_i - p_k, v(j, k|S) - p_j - p_k]$. Summing the inequalities such that the prices cancel, we get: $v(i, j|S) + v(k|S) > \max[v(i|S) + v(j, k|S), v(j|S) + v(i, k|S)]$. Conversely, if you have a violation of the condition (RGP) for i, j, k, S, let $\phi > 0$ be the value of the violation, i.e.,

$\phi = v(i,j|S) + v(k|S) - \max\{v(i|S) + v(j,k|S), v(j|S) + v(i,k|S)\}$. Now, consider prices $p_t = -\infty$ for $t \in S$, $p_t = \infty$ for $t \notin S \cup \{i,j,k\}$ and $p_i = v(i|S \cup \{j\}) - \frac{1}{2}\phi$, $p_j = v(j|S \cup \{i\}) - \frac{1}{2}\phi$ and $p_k = v(k|S) + v(i,j|S) - v(i|S) - v(j|S) - \phi$. It is straightforward to check that such prices give us a certificate of type (ii). $\square$

Now we explain what we mean by a *local* characterization. Given a valuation function v and two sets $S, R \subseteq [n]$ we can define a restriction $v_{R|S} : 2^R \to \mathbb{R}$ by $v_{R|S}(T) = v(T|S)$. We say that this is a k-restriction if $|R| = k$. Observe that usual properties as monotonicity and submodularity are properties of the restrictions, i.e., a function is monotone iff each 1-restriction is monotone. A function is submodular iff each 2-restriction is submodular. A corollary of Theorem 4.1 is that:

COROLLARY 4.3. *A valuation function satisfies the gross substitutes property iff every 3-restriction satisfied the gross substitutes property.*

5. Discrete Hessian and Tree forms. The (RGP) condition implies a rich combinatorial underlying structure for gross substitutes, which we describe in this section. We start by defining the discrete derivative operator and the discrete Hessian. Given a set function $v : 2^N \to \mathbb{R}$ we define for each $i \in N$ the discrete derivative operator ∂_i which associates v with the set function $\partial_i v : 2^{N \setminus i} \to \mathbb{R}$ given by

$$\partial_i v(S) = v(S \cup i) - v(S)$$

In other words, the discrete derivative is the marginal value $v(i|S)$ we have discussed so far. Defining it as an operator allows us to define higher order derivatives by applying it multiple times. For $i \neq j$ we define the second order derivative as:

$$\partial_{ij} v(S) = \partial_i[\partial_j v(S)] = v(S \cup ij) - v(S \cup i) - v(S \cup j) + v(S)$$

We can see from the symmetry in the expression that $\partial_{ij} v = \partial_{ji} v$. This is also true for any higher order derivatives. Even though for the purposes of this paper we are only concerned with second order we briefly mention that for a strict subset $I \subsetneq N$ of items, we can define the derivative $\partial_I v : 2^{N \setminus I} \to \mathbb{R}$ as a successive application of the derivatives. It is not hard to see that the order in which they are applied does not matter and that:

$$\partial_I v(S) = (-1)^{|I|} \sum_{X \subseteq I} (-1)^{|X|} v(S \cup X)$$

Fixed a set S we can define the discrete Hessian as the matrix of values

$$\mathbf{H}[v](S) = [\partial_{ij} v(S)]_{i,j \notin S}$$

The discrete Hessian is not properly a matrix since the diagonal entries are not defined. Let's for now ignore this fact and study the relationship between the symbols $\partial_{ij} v(S)$ for $i, j \notin S$. In the language of discrete derivatives we can rephrase Theorem 4.1 as saying that gross substitutes is equivalent to:

$$\partial_{ij} v(S) \leq \max[\partial_{ik} v(S), \partial_{kj} v(S)] \leq 0 \qquad \text{(Diff)}$$

Where the first inequality is obtained from the (RGP) condition by subtracting $v(i|S) + v(j|S) + v(k|S)$ from each term. The second inequality is equivalent to submodularity.

Permuting the indices if necessary, we can assume that $\partial_{ij}v(S) \geq \partial_{ik}v(S) \geq \partial_{jk}v(S)$. Then (Diff) implies that $\partial_{ij}v(S) \geq \partial_{ik}v(S)$. This is often referred as the isosceles triangle property: for any S, i, j, k up to a permutation of i, j, k we have:

$$0 \geq \partial_{ij}v(S) = \partial_{ik}v(S) \geq \partial_{jk}v(S) \tag{Iso}$$

Bing, Lehmann and Milgrom [9] use the second order derivatives to define a metric on the space of items. Given S we can define a metric $d_S(i,j) = -1/\partial_{ij}v(S)$ over $N \setminus S$. Condition (Diff) is equivalent to saying that the metric d_S is an ultra-metric. An ultra-metric is a metric satisfying a property similar to the triangle inequality with the sum replaced by a maximum:

$$d_S(i,j) \leq \max\{d_S(i,k), d_S(j,k)\}$$

Ultra-metrics have a nice combinatorial description in terms of tree metrics:

THEOREM 5.1 (Murota and Hirai [24]). *A valuation function v satisfies the gross substitutes property iff for any set S there exists a tree T_S having the elements of $N \setminus S$ as the leaves and non-positive labels on the internal nodes such that:*

- *the label of an internal node is smaller than the label of its parent in the tree.*
- *given $i, j \in N \setminus S$, then $\partial_{ij}v(S)$ corresponds to the label of the least common ancestor of i and j in the tree.*

The theorem can be obtained by recursively applying the following lemma:

LEMMA 5.2. *Given a function v satisfying gross substitutes, a set S and $m = \max_{i \neq j, i,j \notin S} \partial_{ij}v(S)$, then there is a partition of S into sets $X_1, \ldots, X_k$ such that:*

- *for $i \in X_a$ and $j \in X_b$ with $a \neq b$, then $m = \partial_{ij}v(S)$.*
- *for $i, j \in X_a$ then $m > \partial_{ij}v(S)$.*

Proof. Define a graph on $N \setminus S$ by adding an edge (i,j) between two items iff $m > \partial_{ij}v(S)$ and take $X_1, \ldots, X_k$ to be the connected components of this graph. First note that if (i,j) and (j,k) are in the graph then (i,k) must be in the graph since: $\partial_{ik}v(S) \leq \max[\partial_{ij}v(S), \partial_{jk}v(S)] < m$, therefore the graph induced on each connected component must be the complete graph. By definition all the edges not in the graph must have $\partial_{ij}v(S) = m$. □

In Figure 5.1, we depict a one step application of the previous lemma. If we draw a node labelled with m in the root and have each child correspond to the set X_i, then the least common ancestor of nodes in different subsets will correspond to the root and will indicate that $\partial_{ij}v(S) = m$ for such nodes. If a set X_i has more than one element, we can apply Lemma 5.2 recursively by setting: $m_1 = \max_{i \neq j, i,j \in X_1} \partial_{ij}v(S)$ and using the same idea to partition X_1 into $Y_1, \ldots, Y_{k_1}$, as depicted in Figure 5.2. By recursively applying this procedure we get to the object described in Theorem 5.1.

We observe one interesting non-trivial and useful consequence of the isosceles triangle property, which will be useful later:

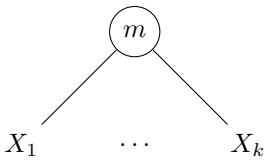

FIG. 5.1. *Example of a one step application of Lemma 5.2*

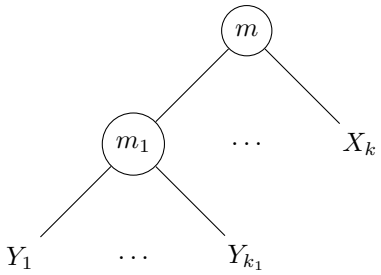

FIG. 5.2. *Example of a two step application of Lemma 5.2*

LEMMA 5.3. *Given a gross substitute valuation* $v : 2^{[n]} \to \mathbb{R}$, $S \subseteq [n]$ *and* $i_1, i_2, j_1, j_2 \notin S$, *then:*

$$\partial_{i_1 i_2} v(S) + \partial_{j_1 j_2} v(S) \leq \max[\partial_{i_1 j_2} v(S) + \partial_{j_1 i_2} v(S), \partial_{i_1 j_1} v(S) + \partial_{i_2 j_2} v(S)]$$

or equivalently,

$$v(i_1, i_2|S) + v(j_1, j_2|S) \leq \max[v(i_1, j_2|S) + v(j_1, i_2|S), v(i_1, j_1|S) + v(i_2, j_2|S)]$$

Proof. There are two possible trees induced on $\{i_1, i_2, j_1, j_2\}$ which are represented in Figure 5.3. Proving the statement of the lemma in each of the two cases is simple: in the first case, we can assume (swapping the names of i_1, i_2 if necessary) that $\partial_{i_1 j_1} v(S) = \partial_{i_1 j_2} v(S) = m_0$, so: $\partial_{i_1 j_1} v(S) + \partial_{i_1 j_2} v(S) = m_0$ which is the maximum achievable value for any pair. In the second case, say i_1 corresponds to node x_1 in the figure, so $\partial_{i_1 x} v(S) = m_0$ for all $x \in \{i_2, j_1, j_2\}$. Therefore we can remove all terms with i_1 from the inequality and the statement in the lemma reduces to condition (Diff).
□

We can use the tree-form of the Hessian (Theorem 5.1) to analyze its spectrum. So far, $\mathbf{H}(S)$ is not a proper matrix since we haven't defined the values on the diagonal. A reasonable way to complete the diagonal is to apply the formula $v(S \cup ij) - v(S \cup i) - v(S \cup j) + v(S)$ used for the other entries when $i = j$ and the fact that $v(S \cup ii) = v(S \cup i)$. Then we obtain $\mathbf{H}_{ii}(S) = -\partial_i v(S)$. Under this definition the matrix becomes negative semidefinite:

LEMMA 5.4 (Dughmi, Roughgarden and Yan [16]). *If we* v *is a non-decreasing function and the diagonal is defined as above, then the Hessian is negative semi-definite, i.e., for all vectors* $u \in \mathbb{R}^{N \setminus S}$, $u^\top \mathbf{H}(S) u \leq 0$.

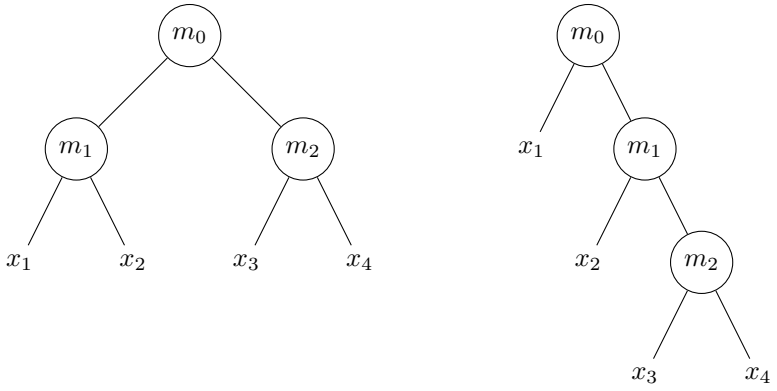

FIG. 5.3. *Two possible trees induced on 4 elements, with $m_0 > m_1 \geq m_2$.*

More generally we will show that every matrix in *negative-tree-form* is negative semi-definite. We define this concept recursively: a 1×1 matrix M is in negative-tree-form if $M_{11} \leq 0$. For an $n \times n$ matrix we say that a matrix is in negative-tree-form if $M_{ij} \leq 0$ for all i, j and there is a subset $\emptyset \subsetneq S \subsetneq [n]$ and a constant m_0 such that for all $i \in S$ and $j \notin S$

$$m_0 = H_{ij} = H_{ji} \text{ and } H_{ii} \leq m_0 \text{ and } H_{jj} \leq m_0$$

and the matrices $M_{S,S} - m_0$ and $M_{N\setminus S, N\setminus S} - m_0$ are in *negative-tree-form*, where $M_{S,S}$ is the submatrix formed by rows and columns indexed by S and $M_{S,S} - m_0$ is the matrix obtained by subtracting each component by m_0.

It follows directly from Theorem 5.1 and from monotonicity that the Hessian of a monotone gross substitutes valuation is in negative-tree-form. Monotonicity is used to argue that $\partial_{ij} v(S) \leq -\partial_i v(S)$. The proof of Lemma 5.4 is a consequence of the following statement:

LEMMA 5.5. *Every matrix is negative-tree-form is negative semidefinite.*

Proof. We again prove it by induction. The 1×1 is trivial. For $n \times n$ matrix, pick m_0 and S as in the recursive definition, then:

$$u^\top M u = m_0(\mathbf{1}^\top u)^2 + u_S^\top (M_{S,S} - m_0) u_S + u_{N\setminus S}^\top (M_{N\setminus S, N\setminus S} - m_0) u_{N\setminus S} \leq 0$$

since $m_0 \leq 0$ and the other terms are non-positive by induction. □

6. Well Layered and Matroidal Maps. The final step towards proving the equivalence of definitions 3.1 and 1.3 is the concept of *well layered maps* introduced by Dress and Terhalle [13] – in which the authors characterize the set functions $v : 2^{[n]} \to \mathbb{R}$ for which greedy algorithms are optimal.

DEFINITION 6.1 (well-layered map). *A function $v : 2^{[n]} \to \mathbb{R}$ is called well-layered iff for each $p \in \mathbb{R}^n$ the sets $S_0, S_1, S_2, \ldots$ obtained by the greedy algorithm (i.e., $S_0 = \emptyset$ and $S_i = S_{i-1} \cup \{x_i\}$ where $x_i \in argmax_{x \in [n]\setminus S_{i-1}} v_p(x|S_{i-1})$) are such that $v_p(S_i) = \max\{v_p(S); |S| = i\}$.*

THEOREM 6.2 (Dress and Terhalle [13]). *A map $v : 2^{[n]} \to \mathbb{R}$ is well-layered iff for any triple of disjoint sets $S, \{i\}, T$ with $|T| \geq 2$,*

$$v(T|S) + v(i|S) \leq \max_{j \in T} v(j|S) + v(T \cup i \setminus j|S) \tag{WL}$$

Before we proceed to the proof of Theorem 6.2 it is useful to notice its relation with the condition of Reijnierse et al:

LEMMA 6.3. *Conditions (RGP) and (WL) are equivalent.*

Proof. One readily recognizes condition (RGP) to be a special case of (WL) with $|T| = 2$, i.e., $T = \{j, k\}$. For the other direction, we show by induction on $|T|$ that (RGP) implies (WL). For $|T| = 2$, this is trivial. Now, suppose we proved it for $|T| = t - 1$. Given $S, \{i\}, T$ with $|T| = t$, choose $k \in T$ minimizing $\alpha_S(i, k)$. Then by the induction hypothesis applied to $S \cup k, \{i\}, T \setminus k$, we have that there is $j \in T \setminus k$ such that:

$$v(i|S \cup k) + v(T \setminus k|S \cup k) \leq v(j|S \cup k) + v(T \cup i \setminus j, k|S \cup k)$$

which can be re-written as:

$$v(i, k|S) + v(T|S) \leq v(j, k|S) + v(T \cup i \setminus j|S)$$

now notice that by the choice of k, it must be the case that for all j, $\alpha_S(i, k) \leq \alpha_S(k, j)$, since the smaller α_S-value in the triangle i, j, k appears twice and $\alpha_S(i, k) \leq \alpha_S(i, j)$. Now, this means that $v(i|S) - v(i, k|S) \leq v(k|S) - v(j, k|S)$. Summing with the previous inequality gives us condition (WL) for $|T| = t$. □

Proof of Theorem 6.2. First, assume that the condition (WL) holds and let's prove that $S_t \in \operatorname{argmax}_{S;|S|=t} v_p(S)$ by induction on t. The case $t = 1$ is trivial. Assume we proved for $t - 1$ and assume there is S' with $|S'| = t$ and $v_p(S') > v_p(S_t)$. Choose such set maximizing k such that $\{x_1, \ldots, x_k\} \subseteq S'$. Since $|S'| = |S_t| = t$, clearly $k < t$. Now, applying (WL) for $\{x_1, \ldots, x_k\}, x_{k+1}, T' = S' \setminus \{x_1, \ldots, x_k\}$ we get that there is $j \in T'$ such that:

$$v(x_{k+1}|x_1, \ldots, x_k) + v(T'|x_1, \ldots, x_k) \leq v(j|x_1, \ldots, x_k) + v(T' \cup x_{k+1} \setminus j|x_1, \ldots, x_k)$$

and since $v(x_{k+1}|x_1, \ldots, x_k) \geq v(j|x_1, \ldots, x_k)$ by the greedy rule, we have that $v(S') \leq v(S' \cup x_{k+1} \setminus j)$ contradicting the minimality of k.

For the other direction, assume that (WL) is violated. Since (WL) is equivalent to (RGP), (WL) must be violated by some $|T| = \{j, k\}$. So assume $S, i, \{j, k\}$ for which (WL) is not valid. Now, define prices such that $p_t = -\infty$ for $t \in S$, $p_t = \infty$ for $t \notin S \cup \{i, j, k\}$, $p_i = v(i|S) - \epsilon$, $p_j = v(j|S)$ and $p_k = v(k|S)$. The greedy algorithm will first pick all the elements in S, then i. Now, observe that for $t = |S| + 2$ the optimal set is either $S \cup \{i, j\}$, or $S \cup \{j, k\}$ or $S \cup \{i, k\}$. The fact that (WL) is violated implies that $v(i|S) + v(j, k|S) > v(j|S) + v(i, k|S)$. Substituting $v(i|S)$ and $v(j|S)$ by the prices, we get that (for sufficiently small ϵ) $v(j, k|S) - p_j - p_k > v(i, k|S) - p_i - p_k$, i.e., $S \cup \{j, k\}$ is strictly preferable then $S \cup \{i, k\}$. The exact same argument works swapping j and k, so the only set of size $|S| + 2$ maximizing v_p is $S \cup \{j, k\}$. Therefore v can't be a well-layered map, since the greedy algorithm picked i in step $|S| + 1$. This

finishes the proof of Theorem 6.2. □

The concept of well layered maps guarantees that the greedy algorithm will find the optimal set of each cardinality. In order to guarantee that the greedy algorithm as described in Section 3 will find the optimal, we need to guarantee that once a layer doesn't improve over the previous, we can stop. Dress and Terhalle [12] observe that in order for this to happen, it is necessary and sufficient that the valuation is both well-layered and submodular. They call such functions *matroidal maps*.

THEOREM 6.4 (Dress and Terhalle [12]). *A valuation function satisfied Definition 3.1 iff it is well-layered and submodular.*

Proof. Submodularity guarantees that the sets S_t will be such that $v_p(S_{t+1}) - v_p(S_t) \leq v_p(S_t \cup x_{t+1} \setminus x_t) - v_p(S_{t-1}) \leq v_p(S_t) - v_p(S_{t-1})$. So: $v_p(S_t) \geq \frac{1}{2}[v_p(S_{t-1}) + v_p(S_{t+1})]$. This guarantees that $t \mapsto v_p(S_t)$ is concave. For the converse, if a function is not submodular, then there exist i, j, S such that $v(i,j|S) > v(i|S) + v(j|S)$. So one can set prices $p_t = -\infty$ for $t \in S$, $p_t = \infty$ for $t \notin S \cup \{i,j\}$, $p_i = v(i|S) + \epsilon$ and $p_j = v(j|S) + \epsilon$ for some small ϵ. For such prices the greedy algorithm will terminate on S, while the optimum is $S \cup \{i,j\}$. □

Theorem 6.4 establishes our main claim that Definitions 3.1 and 1.3 are equivalent. In particular, for any gross substitute valuation function, a set $S \in D(v;p)$ can be found using the greedy algorithm (Algorithm 2). The following remark states that the greedy algorithms finds not only one, but all demanded sets. This fact is somehow implicit in the proof of Theorem 6.4 but we state and prove it again for completeness:

REMARK 6.5. *Given a gross substitute valuation function v, a price vector p and a demanded set $S \in D(v;p)$, then there is a tie breaking rule*[4] *for the greedy algorithm that produces the set S. Formally, if $x_1, x_2, \ldots, x_k$ is an ordering of elements in S such that $x_i \in argmax_{y \in S \setminus \{x_1,\ldots,x_{i-1}\}} v(y|x_1,\ldots,x_{i-1})$, then x_i is also in $argmax_{y \in [n] \setminus \{x_1,\ldots,x_{i-1}\}} v(y|x_1,\ldots,x_{i-1})$.*

Proof. This fact follows directly from the (WL) condition. Assume that there is i such that for some $v_p(y|x_1,\ldots,x_{i-1}) > v_p(x_i|x_1,\ldots,x_{i-1})$. Since $x_1,\ldots,x_k$ are in greedy order, it must be that $y \notin S$. By applying (WL) with sets $X_{i-1} = \{x_1,\ldots,x_{i-1}\}, \{y\}, \{x_i,\ldots,x_k\}$, we get that there is x_t for $t \geq i$ such that:

$$v_p(y|X_{i-1}) + v_p(x_i,\ldots,x_k|X_{i-1}) \leq v_p(x_t|X_{i-1}) + v_p(\{y,x_i,\ldots,x_k\} \setminus x_t|X_{i-1})$$

Since $v_p(y|X_{i-1}) > v_p(x_i|X_{i-1})$, we get $v_p(S) < v_p(S \cup y \setminus x_t)$, which contradicts that $S \in D(v;p)$. □

We would like to finish by pointing out that many combinatorial structures such as matroids, polymatroids, valuated matroids [14], among others, can be defined as the class of objects for which a certain problem admits a greedy solution. For a more extensive exposition on such combinatorial structures we refer to the classical text of Korte, Lovász and Schrader on *greedoids* [28]. Similar arguments can be used to

[4] formally a *tie breaking rule* is a function b that associates for each pair of disjoint sets S, T an element from T. The greedy algorithm with tie-breaking rule b, whenever choosing an element in $T = \text{argmax}_{i \notin S} v(i|S)$, it adds $b(S,T) \in T$.

argue about local search:

THEOREM 6.6. *A valuation function satisfied Definition 3.3 iff it is well-layered and submodular.*

Proof. First we argue that if a valuation v is well-layered and submodular, then local search can't get stuck in a local maximum for any price p. Let $S^* \in \text{argmax}_{S\subseteq[n]} v_p(S)$ and $S \subseteq [n]$ be such that $v_p(S) < v_p(S^*)$. Then we want to argue that there is $S' \in \mathcal{N}$ where $\mathcal{N}$ is the neighborhood of S as in the local search procedure in Section 3. First observe that if v is well-layered and submodular, then v_p has also those two properties.

We consider three cases:

Case (i) $S \subseteq S^*$. Notice that $0 < v_p(S^* \setminus S|S) \leq \sum_{i\in S^*\setminus S} v_p(i|S)$, where the last inequality follows from submodularity. Therefore, there is some i for which $v_p(i|S) > 0$, then we can take $S' = S \cup \{i\}$.

Case (ii) $S^* \subseteq S$. Let $S \setminus S^* = \{i_1, \ldots, i_k\}$ For this case, $0 > v_p(S \setminus S^*|S^*) = \sum_{j=1}^{k} v_p(i_j|S \cup \{i_1, \ldots, i_{j-1}\}) \geq \sum_{j=1}^{k} v_p(i_j|S \setminus i_j)$. So there must be $i \in S \setminus S^*$ such that $v_p(i|S \setminus i) < 0$, then we can take $S' = S \setminus i$.

Case (iii) if neither $S \subseteq S^*$ nor $S^* \subseteq S$, let i be the element in $(S \setminus S^*) \cup (S^* \setminus S)$ maximizing $v_p(i|S \cap S^*)$. Now, we consider two possibilities: (iii-a) If $i \in S^*$, we use condition (WL) with $S \cap S^*, i, S \setminus S^*$. This gives us $j \in S \setminus S^*$ such that:

$$v_p(i|S \cap S^*) + v_p(S \setminus S^*|S \cap S^*) \leq v_p(j|S \cap S^*) + v_p((S \setminus S^*) \cup i \setminus j|S \cap S^*)$$

By the choice of i, we know that $v_p(j|S \cap S^*) \leq v_p(i|S \cap S^*)$. If this inequality holds strictly, then, $v_p(S \setminus S^*|S \cap S^*) < v_p((S \setminus S^*) \cup i \setminus j|S \cap S^*)$ and therefore $v_p(S) \leq v_p(S \cup i \setminus j)$. If on the other hand $v_p(j|S \cap S^*) = v_p(i|S \cap S^*)$ then we can swap i and j and consider the sub-case where $i \in S$. (iii-b) If $i \in S$, we use condition (WL) with $S \cap S^*, i, S^* \setminus S$. Doing as above, we find $j \in S^* \setminus S$ such that $v_p(S^*) \leq v_p(S^* \cup i \setminus j)$, which holds with equality since S^* is optimal. This way we obtain an optimal set closer to S. Then we can repeat the above procedure with $S^* \cup i \setminus j$ instead of S^* a finite number of times until we reach some set $S' \in \mathcal{N}$ or we reach one of the previous cases.

For the converse direction, we want to show that if a valuation is not well-layered or not submodular, local search can get stuck in suboptimal local minima. For this, we can use the same examples used in Lemma 6.4 to show that in such case the greedy algorithm can be suboptimal. □

We end this section with Figure 6.1 summarizing the equivalences proved so far. The dashed arrow corresponds to a claim proved in [44] but not fully proved in this survey. To make the survey self-contained, we give a direct proof that the concept of matroidal maps (Definition 3.1) implies the original definition of gross substitutes due to Kelso and Crawford (Definition 1.3):

THEOREM 6.7. *Every matroidal map satisfied the gross substitutes condition.*

Proof. Consider a matroidal map v, a price vector p and a set $S \in D(v; p)$. In order to show that the conditions in Definition 1.3 hold, we only need to show for a price vector p' such that $p'_i > p_i$ for some i and $p'_j = p_j$ for all $j \neq i$, since we can increase prices one at a time. Let S' be a set in $\text{argmax}_{S'\in[n]\setminus i} v_p(S')$. Clearly, if

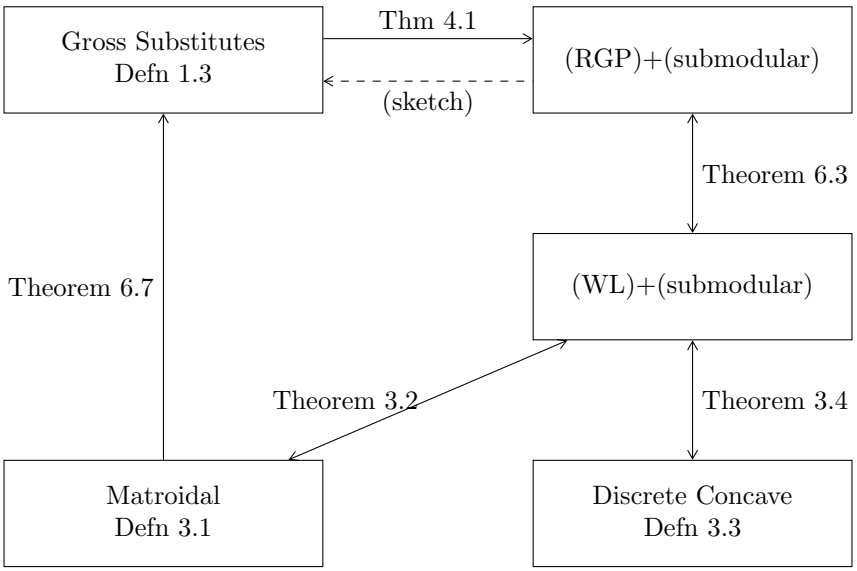

FIG. 6.1. *Summary of equivalences proved so far*

$p'_i \leq p_i + v_p(S) - v_p(S')$, then $S \in D(v;p')$ and we are done. Now, consider the case where $p'_i > p_i + v_p(S) - v_p(S')$. For this, first define $\tilde{p}$ such that $\tilde{p}_i = p_i + v_p(S) - v_p(S')$ and $\tilde{p}_j = p_j$ for all $j \neq i$. Clearly $S \in D(v;\tilde{p})$. By Remark 6.5, there is a tie breaking rule for which the greedy algorithm picks S. Since $v_{\tilde{p}}(i|S \setminus i) = 0$, we can assume that i is the last element picked under this tie breaking rule. One can use the same tie breaking rule to pick under price vector p'. It is straightforward to see that the greedy algorithm behaves the same way as in $\tilde{p}$ until i. Therefore it will choose a set containing $S \setminus i$. □

7. Fenchel Duality for gross substitutes. The duality between gross substitutes and submodular functions was observed in many places, as in Fujishige and Yang [19] and Murota [33], Gul and Stacchetti [22] and Ausubel and Milgrom [2]. Given a valuation function $v : 2^{[n]} \to \mathbb{R}$, we define its Fenchel-dual as the function $u : \mathbb{R}^n \to \mathbb{R}$ given by: $u(p) = \max_S v(S) - p(S)$ Economically, this corresponds to the utility function mapping prices $p \in \mathbb{R}^n$ to highest possible utility the buyer can obtain by buying a bundle of items under the current prices.

They relate it to the concept of $\mathbb{R}^n$-valued submodular function, which are functions $f : \mathbb{R}^n \to \mathbb{R}$ such that for any $x, y \in \mathbb{R}^n$, $f(x \vee y) + f(x \wedge y) \leq f(x) + f(y)$, where $\vee$ and $\wedge$ are the componentwise maximum and minimum respectively. Analogously to submodular set functions, this is equivalent to $f(x + \delta_i \cdot e^i + \delta_j \cdot e^j) - f(x + \delta_i \cdot e^i) \leq f(x + \delta_j \cdot e^j) - f(x)$ for any $\delta_i, \delta_j > 0$ and $i \neq j$, where e_i is the i-th coordinate vector.

THEOREM 7.1 (duality, Ausubel and Milgrom [2]). *A valuation function $v : 2^{[n]} \to \mathbb{R}$ has the gross substitutes property iff its associate Fenchel-dual $u : \mathbb{R}^n \to \mathbb{R}$ is an $\mathbb{R}^n$-valued submodular function.*

Proof. Since u is continuous, it is enough to prove for almost all p, δ_i, δ_j and then we can extend to all by continuity. Define $\Gamma = \cup_{S,T \subseteq [n]} \{p \in \mathbb{R}^n; v_p(S) = v_p(T)\}$. Then Γ is a measure zero subset of $\mathbb{R}^n$, since it is a finite collection of hyperplanes. For $p \notin \Gamma$,

$|D(v,p)| = 1$. For $p \notin \Gamma$, denote by $D(v,p)$ the unique set demanded at those prices. Given such p, there is ball around p such that for all price vectors, the demand set is the same, so $u(p) = v(D(v,p)) - p(D(v,p))$ and therefore, $\frac{d}{dp_j}u(p) = -\mathbb{1}\{j \in D(v,p)\}$. Now, notice that given p, δ_i, δ_j:

$$[u(p + \delta_i \cdot e^i + \delta_j \cdot e^j) - u(p + \delta_i \cdot e^i)] - [u(p + \delta_j \cdot e^j) - u(p)] =$$
$$\int_0^{\delta_j} \frac{d}{dp_j}u(p + \delta_i \cdot e^i + z \cdot e^j) - \frac{d}{dp_j}u(p + z \cdot e^j)dz =$$
$$\int_0^{\delta_j} -\mathbb{1}\{j \in D(v, p + \delta_i \cdot e^i + z \cdot e^j)\} + \mathbb{1}\{j \in D(v, p + z \cdot e^j)\}dz \leq 0$$

since the increase in p_i can't remove j from the demand set by gross substitutability. The converse can be proved by the same argument backwards.
□

A different view of the same duality can be obtained by the characterization of demand sets as basis of a matroid:

THEOREM 7.2 (duality, Gul and Stacchetti [22]). *A valuation function $v : 2^{[n]} \to \mathbb{R}$ has the gross substitutes property then for any price $p \in \mathbb{R}^n$, the set*

$$D^*(v,p) = \{S \in D(v,p); |S| \leq |T|, \forall T \in D(v,p)\}$$

of the demanded sets of minimum size, form the set of basis of a matroid.

One characterization of basis-set $\mathcal{B} \subset 2^{[n]}$ of matroids is via the exchange property: for any $S, T \in \mathcal{B}$ and $s \in S \setminus T$, there is $t \in T \setminus S$ such that $S \cup t \setminus s \in \mathcal{B}$. We notice that we proved exactly this fact in case (iii) of the proof of Theorem 6.6.

8. Connection to Discrete Convex Analysis and Valuated Matroids. Fujishige and Yang [19] showed a powerful connection between gross substitute valuations and the concept of $M^\natural$-concave functions in Discrete Convex Analysis. Discrete Convex Analysis is a theory developed by Murota [33] that defines a very general class of functions $f : \mathbb{Z}^n \to \mathbb{R}$ on the integral lattice for which it is possible to prove strong duality theorems. Such theorems enable the design of efficient greedy and flow-like algorithmic solutions for various discrete optimization problems involving such functions.

Murota and Shioura [39] define $M^\natural$-concave functions based on the concept of M-concavity of Murota [33]. They originally define $M^\natural$-concave functions on the integral lattice $\mathbb{Z}^n$, but for the purposes of this survey, we will consider their restriction to $\{0,1\}^n$:

DEFINITION 8.1 ($M^\natural$-concave functions, Murota and Shioura [39]). *A function $v : 2^{[n]} \to \mathbb{R}$ is $M^\natural$-concave if for all $S, T \subseteq [n]$ and $s \in S \setminus T$,*

$$v(S) + v(T) \leq \max\left[v(S \setminus s) + v(T \cup s), \max_{t \in T \setminus S} v(S \cup t \setminus s) + v(T \cup s \setminus t)\right] \qquad (M^\natural)$$

THEOREM 8.2 (Fujishige and Yang [19]). *A function $v : 2^{[n]} \to \mathbb{R}$ has the gross substitutes property iff it is $M^\natural$-concave.*

Proof. The fact that $M^\natural$-concavity implies gross substitutability is easy to see, since taking $T \subseteq S \setminus s$ we recover submodularity: $v(S) + v(T) \leq v(S \setminus s) + v(T \cup s)$ which can be rewritten as $v(s|S \setminus s) \leq v(s|T)$. Taking $|S \setminus T| = 1$ and $|T \setminus S| \geq 2$, we recover (WL). Since by submodularity $v(S) + v(T) \geq v(S \setminus s) + v(T \cup s)$, so if $v(S)+v(T) \leq v(S\setminus s)+v(T\cup s)$, then $v(U\cup s) = v(U)+v(s|T)$ for any $S \subseteq U \subseteq T$, making (WL) hold for any $t \in T\setminus S$. On the other hand, if $v(S)+v(T) > v(S\setminus s)+v(T\cup s)$, then: $v(S) + v(T) \leq \max_{t\in T\setminus S} v(S \cup t \setminus s) + v(T \cup s \setminus t)$, which is exactly (WL).

For the other direction, assume that v is gross substitutes and we want to show it satisfies $(M^\natural)$. First, consider the following transformation: given $v : 2^{[n]} \to \mathbb{R}$, define another valuation function on $2n$ items by adding n dummy items: $\omega : 2^{[2n]} \to \mathbb{R}$, $\omega(S) = v(S \cap [n])$. It is straightforward to check that if v is gross substitutes then so is ω. Now we define the condition (M) on ω as follows: for all sets $S, T \subseteq [2n]$ with $|S| = |T|$ and $s \in S \setminus T$:

$$\omega(S) + \omega(T) \leq \max_{t\in T\setminus S} \omega(S \cup t \setminus s) + \omega(T \cup s \setminus t) \tag{M}$$

It is simple to see that if ω satisfied (M) for all sets of equal cardinality, then v satisfied $(M^\natural)$, since any pair of sets $S, T \subseteq [n]$ map to equal cardinality sets $S', T' \subseteq [2n]$ by padding the smaller set with dummy elements. Notice that (M) on ω implies $(M^\natural)$ on v, since the term $v(S \setminus s) + v(T \cup s)$ accounts for the possibility that $t \in T \setminus S$ in (M) is a dummy item of $[2n]$.

So, we only need to prove that if ω is a gross substitutes valuation, then it also satisfied (M). For $|S \setminus T| = |T \setminus S| = 1$, the property in trivial. For $|S \setminus T| = |T \setminus S| = 2$, this follows directly from Lemma 5.3. Now we prove by induction on $k = |S \setminus T| = |T \setminus S|$. Fix some arbitrary $\tilde{s} \in S \setminus (T \cup s)$ and find $\tilde{t} \in T \setminus S$ maximizing $\omega(T\cup\tilde{s}\setminus\tilde{t})-\omega(S\cup\tilde{t}\setminus s)$. Now, apply induction on the sets S and $T\cup\tilde{s}\setminus\tilde{t}$. We get that there is $t \in T \setminus (S \cup \tilde{t})$ such that:

$$\omega(S) + \omega(T \cup \tilde{s} \setminus \tilde{t}) \leq \omega(S \cup t \setminus s) + \omega(T \cup \{s, \tilde{s}\} \setminus \{t, \tilde{t}\})$$

By the case with $k = 2$ with sets T and $T \cup \{s, \tilde{s}\} \setminus \{t, \tilde{t}\}$, we know that:

$$\omega(T)+\omega(T\cup\{s,\tilde{s}\}\setminus\{t,\tilde{t}\}) \leq \max[\omega(T\cup s\setminus t)+\omega(T\cup\tilde{s}\setminus\tilde{t}),\omega(T\cup\tilde{s}\setminus t)+\omega(T\cup s\setminus\tilde{t})]$$

If the maximum corresponds to the first expression, this together with the previous inequality, gives us exactly what we want to prove, i.e., $\omega(S)+\omega(T) \leq \omega(S\cup t\setminus s)+\omega(T\cup s\setminus t)$, which corresponds to condition (M). If the maximum is the second expression we use the choice of $\tilde{t}$ to see that: $\omega(T\cup\tilde{s}\setminus\tilde{t})-\omega(S\cup\tilde{t}\setminus s) \geq \omega(T\cup\tilde{s}\setminus t)-\omega(S\cup t\setminus s)$. This together with the previous inequalities also leads to condition (M). □

An important consequence of this equivalence is that it is possible to obtain a geometric characterization of gross substitutes in terms of $M^\natural$-convex sets. Murota defines an analogue of convex sets in the lattice $\mathbb{Z}^n$ (which in capture the notion that the sets have no 'holes') and shows that a function is gross substitutes if and only if for every price vector p, the demanded $D(v; p)$ is an $M^\natural$-convex set. Although more geometrically appealing in the $\mathbb{Z}^n$-lattice, we give below the definition of $M^\natural$-convex set restricted to the hypercube $\{0, 1\}^n$.

Definition 8.3 ($M^\natural$-convex set). *A subset $X \subseteq \{0, 1\}^n$ is $M^\natural$-convex set if for any two sets $S, T \in X$ and element $i \in S \setminus T$ then either (i) $S \setminus i$ and $T \cup i$ are in X*

or: (ii) there is an element j such that $S \cup j \setminus i$ and $T \cup i \setminus j$ are in X.

This geometric characterization implies the characterization in Theorem 7.2, since it is easy to see from the definition that the sets in X of minimum cardinality form the set of basis of a matroid. The same is true for the set of maximum cardinality.

Valuated Matroids. The characterization of gross substitutability by the ($M^\natural$) also connects it to the concept of *valuated matroids*, due to Dress and Wenzel [14]:

DEFINITION 8.4. *Let $\binom{[n]}{k} = \{S \subseteq [n]; |S| = k\}$. We say that a map $\omega : \binom{[n]}{k} \to \mathbb{R}$ is a* valuated matroid *if it satisfies the following version of the exchange property: given $S, T \in \binom{[n]}{k}$ and $s \in S \setminus T$, there exists $t \in T \setminus S$ such that:*

$$\omega(S) + \omega(T) \leq \omega(S \cup t \setminus s) + \omega(T \cup s \setminus t)$$

In particular, Theorem 8.2 together with the discussion in its proof imply that:

LEMMA 8.5. *A valuation $v : 2^{[n]} \to \mathbb{R}$ satisfies the gross substitutes property iff the map $\omega : \binom{[2n]}{n} \to \mathbb{R}$ defined by $\omega(S) = v(S \cap [n])$ is a valuated matroid. Also, if v satisfies the gross substitutes property then for every $k \leq n$, the restriction of v to $\binom{[n]}{k}$ is a valuated matroid. In other words, given two sets of equal cardinality and a gross substitutes valuation, then (M) is satisfied.*

9. Convolution operation. As we saw in Section 2, one *cannot* build gross substitute functions by taking linear combinations of simpler gross substitute, since gross substitutability is not closed under addition. However, this class is closed under a different operation, called convolution.

THEOREM 9.1 (Lehmann, Lehmann and Nisan [31] and Murota [33]). *Given two valuation functions v^1, v^2 satisfying the gross substitutes property then the valuation function $v = v^1 * v^2$ also satisfied the gross substitutes property, where:*

$$v^1 * v^2(S) = \max_{S_1 \subseteq S} v^1(S_1) + v^2(S \setminus S_1)$$

Proof. We will give an algorithmic proof of the previous theorem based on a couple of observations about the Walrasian tatônnement procedure. First note that we can find $S \in D(v^1 * v^2, p)$ by finding a Walrasian equilibrium in an economy with items $[n]$ and three players with valuations v^1, v^2, u where $u(S) = \sum_{j \in S} p_j$. Let S_1, S_2, U be the partition of the items induced by such equilibrium. Then, this is the partition maximizing $v_1(S_1) + v_2(S_2) + p(U) = p([n]) + [v_1(S_1) + v_2(S_2) - p(S_1 \cup S_2)]$. In particular, S_1, S_2 must be the optimal partition of $S = S_1 \cup S_2$ among v^1, v^2, therefore, S maximizes $(v^1 * v^2)(S) - p(S)$ and therefore $S \in D(v^1 * v^2, p)$.

Second observe that for gross substitute valuations the Walrasian tatônnement procedure (Algorithm 1) always outputs a partition of the goods if we are careful to always select $X_i \in D(v^i, p^i)$ such that $S_i \subseteq X_i$. This can be easily implemented by computing X_i via the greedy algorithm (Algorithm 2) initialized with $X_i = S_i$. Moreover, the partition is such that $\sum_i v^i(S_i) \geq \sum_i v^i(S_i^*) + \delta n$. For rational valuations, we can rescale them such that $v^i(S_i)$ are integers. In such case, taking $\delta < \frac{1}{n}$ guarantees that Walrasian tatônnement outputs the optimal allocation.

Finally, in the description of Algorithm 1 we initialized the prices as zero and the allocations such that agent 1 initially has all the goods. Notice that it enough to initialize with a price $p \in \mathbb{R}^n_+$ and a partition $S_1, \ldots, S_n$ such that there is $X_i \in D(v^i, p)$ such that $S_i \subseteq X_i$.

Those observations together can be used to give an elementary proof of Theorem 9.1. We show that $v^1 * v^2$ satisfy the Definition 1.3. Let $S \in D(v^1 * v^2, p)$ and $(v^1 * v^2)(S) = v^1(S_1) + v^2(S_2)$. Consider a Walrasian equilibrium in the economy formed by v^1, v^2, u. Let q be the price vector in such equilibrium. By the Second Welfare Theorem, we take the allocation in equilibrium as $S_1, S_2, U = [n] \setminus (S_1 \cup S_2)$. Let p' be a price vector with $p \leq p'$. We want to show that there is a set $X \in D(v^1 * v^2, p')$ such that $S \cap \{j; p_j = p'_j\} \subseteq X$.

For that, define $u'(S) = \sum_{j \in S} p'_j$ and consider the Walrasian tatônnement procedure for the economy defined by v^1, v^2, u' . Initialize such procedure with allocation S_1, S_2, U and price vector q. This is a valid initialization, since $S_i \in D(v^i, q)$ and also, $U \subseteq \{j; q_j \leq p'_j\} \in D(u', q)$. Now, let S'_1, S'_2, U' be the final outcome of the Walrasian tatônnement procedure. Observe that if $j \in S_1 \cup S_2$, then $q_j \geq p_j$, otherwise q wouldn't be Walrasian for v^1, v^2, u. Now, if $p'_j = p_j$, then such item couldn't have been acquired by u', since it would never be in his demand for such price. Therefore $j \in S'_1 \cup S'_2$. Hence, $(S_1 \cup S_2) \cap \{j; p_j = p'_j\} \subseteq S'_1 \cup S'_2$. $\square$

This is somewhat remarkable that gross substitutability is preserved under convolution. The same is not true for submodularity, for example. The following example is due to Lehmann, Lehmann and Nisan [31]: consider two additive functions v^1 and u over three items given by the vectors $[1, 2, 0]$ and $[3, 5, 3]$ respectively. Now, define $v^2(S) = \min(6, u(S))$. Then both v^1 and v^2 are submodular, but the convolution $v = v^1 * v^2$ is not, since:

$$v(1, 2) + v(2, 3) = 6 + 6 < 8 + 5 = v(1, 2, 3) + v(2)$$

A corollary of Theorem 9.1 is that assignment valuations are gross substitutes: it is straightforward from the definition that an assignment valuation function can be written as a convolution of unit-demand functions, one for each right-side node in the bipartite graph.

10. Computing Walrasian Prices for gross substitutes. The problem of computing a Walrasian equilibrium of an economy consisting of n items and m agents with gross substitutes valuations $v^1, \ldots, v^m$ has two components: the first is called the *welfare problem*, which consists of finding a partition $S_1, \ldots, S_m$ maximizing $\sum_i v^i(S_i)$. The second is the computation of *Walrasian prices*. There are various approaches for those problems: the perhaps more classical line of approach is to use variations of the tatônnement procedure. Nisan and Segal [40] propose a solution that explores properties of gross substitutes to build a suitable linear program. Finally, Murota [34, 35] gives a strongly polynomial time algorithm for this problem based on a cycle-cancelling approach.

We start by discussing how to obtain Walrasian prices from a solution to the welfare problem. The first method is based on an idea by Gul and Stachetti [21]:

LEMMA 10.1 (Gul and Stachetti [21]). *Let W be the optimal welfare of an economy with a set $[n]$ of items and agents with gross substitute valuations $v^1, \ldots, v^m$.*

Also, let W_{-j} be the welfare with the economy with the same agents and items $[n] \setminus j$. Then the price vector p with $p_j = W - W_{-j}$ is a vector of Walrasian prices for the original economy.

The method proposed by Gul and Stachetti to compute Walrasian prices needs access to the optimal allocation for $n+1$ economies: the original one and the economy after each good is removed. An alternative approach is given by Murota [34]. First, consider the following definition:

DEFINITION 10.2 (Exchange graph). *Let $S_1, \ldots, S_m$ be an arbitrary allocation of a set $[n]$ of items to agents with gross substitute valuations $v^1, \ldots, v^m$. For convenience, we consider m additional dummy $\phi_1, \ldots, \phi_m$ and extend the valuations to this set in such a way that for each set S, $v^i(S) = v^i(S \cap [n])$. Also, let $S'_i = S_i \cup \phi_i$.*

The exchange graph is defined as a directed weighted graph with nodes $[n+m] = [n] \cup \{\phi_1, \ldots, \phi_m\}$ and weighted directed edges

$$(j,k) \text{ with weight } w_{jk} = -v^i(S_i \cup k \setminus j) + v^i(S_i) \text{ for } j \in S'_i \text{ and } k \notin S'_i$$

The exchange graph represents possible trades between agents and the (negative) value of each individual trade. As we will formalize in the following lemma, a cycle of negative length implies the existence of trades that improve the welfare of the allocation. If there are no negative cycles, we can certify this fact via the node potentials which can be interpreted as Walrasian prices.

LEMMA 10.3 (Murota [34]). *If the allocation is optimal, the exchange graph has no negative cycles and therefore, the shortest-path distance is well defined. For each $i \in [n+m]$, let d_i be the distance from dummy node ϕ_1 to i. Then $d_i \leq 0$ and the vector $p_i = -d_i$ is a vector of Walrasian prices.*

Proof. First assume that the graph has no negative cycles. In such case, the concept of distance is well defined. Given a pair of dummy nodes ϕ_i, ϕ_j, the weight of the arcs $w_{\phi_i,\phi_j} = w_{\phi_j,\phi_i} = 0$, then $d_{\phi_i} = 0$ for all dummy nodes. Also, for all items $k \in S_i$, the weight of the arc from a dummy node ϕ_i to k is $w_{\phi_i,k} = -v^i(S_i \cup k) + v^i(S_i) \leq 0$, so $d_k \leq 0$ for all nodes k. This allows us to define prices as $p_k = -d_k$. Since ϕ is the shortest path distance, for all $j \in S'_i, k \notin S'_i$: $d_k \leq d_j + w_{jk}$, which is equivalent to: $v^i(S_i) \geq v^i(S_i \cup k \setminus j) - p_k + p_j$. Since k and j are possibly dummy items, this also implies that: $v^i(S_i) \geq v^i(S_i \cup k) - p_k$ and $v^i(S_i) \geq v^i(S_i \setminus j) + p_j$. The last three inequalities show that S_i is a local optimal of the local search procedure (Algorithm 3), hence $S_i \in D(v^i, p)$ by Theorem 6.6. Therefore, p is a vector of Walrasian prices.

Now we argue that if the allocation is optimal, then there are no negative cycles. Given an optimal allocation, let p be a vector of Walrasian prices supporting this allocation. Now define $d_j = -p_j$ for all $j \in [n]$ and $d_d = 0$ for all dummy items d. By the same argument as above, the fact that p is a vector of Walrasian prices implies that: $w_{ij} \geq d_i - d_j$, therefore for every cycle $i_1, i_2, \ldots, i_k, i_1$ we have: $\sum_{t=1}^{k} w_{i_t i_{t+1}} \geq \sum_{t=1}^{j} d_{i_t} - d_{i_{t+1}} = 0$. □

The exchange graph has $O(n+m)$ nodes and $O((n+m)^2)$ edges. Checking if a graph has negative cycles can be done in time $O(N \cdot E)$ using the Bellman-Ford

algorithm, where N is the number of nodes and E is the number of edges. This gives an $O((n+m)^3)$ algorithm to compute Walrasian prices for gross substitute valuations given the optimal allocation.

11. Welfare Problem for gross substitutes. Finally, we discuss algorithmic solutions to the welfare problem for gross substitute valuations. Before we start, we mention a couple of important special cases of this problem. If $v^i(S) = \max_{j \in S} w_{ij}$ for all $i \in [m]$, then this is the traditional *maximum weighted matching* problem. If v^i is the rank function of a matroid, then this corresponds to a special case of the *matroid intersection problem*. For example, the problem of deciding if a graph has k disjoint spanning trees naturally maps to the welfare problem with k agents where the items correspond to edges of the graph and valuation functions correspond to the rank function of the graphical matroid. We discuss three approaches for this problem: tatônnement, linear-programming and cycle cancelling. The first approach has a natural economic intuition but yields only an approximation scheme. The second approach produces an exact solution and runs in polynomial time. The third approach is purely combinatorial and yields a strongly polynomial time algorithm.

11.1. Algorithms via the tatônnement procedure. In Section 1, the Walrasian tatônnement procedure (Algorithm 1) was used as a proof device to show the existence of Walrasian equilibria for gross substitute valuations. In this section we discuss how to use it as an actual algorithm. We start by analyzing the running time of Algorithm 1 using the greedy algorithm (initialized with $X_i = S_i$) to compute the demand oracle. Then we discuss variants of the implementation.

We assume that $v^i(S)$ is an integer (rescaling the input, if necessary) and define $M = \max_{i \in [m]} v^i([n])$. We argued in Section 1 that each price can increase at most M/δ times. This gives a bound of nM/δ on the number of total price increases.

In what follows we argue that there are at most $m + nM/\delta$ executions of the *while* loop in Algorithm 1. Consider the following implementation of the while loop: start with a queue containing all the agents $1, \ldots, m$. At each time, pop agent i from the queue and compute $X_i \in D(v^i, p^i)$ with $S_i \subseteq X_i$. If $X_i \neq S_i$, execute the while loop and for each $k \neq i$ such that S_k changes during the while loop, add k to the queue if he is not already there.

Noticed at this point we removed i from the queue. After the execution of the while loop, we don't need to look at i again, *unless* S_i changes, i.e., unless some item j is taken away from i, since by the fact that valuations are gross substitutes and the prices only increase during the process, $S_i \in D(v^i, p^i)$ if the prices of items in S_i stay the same.

Each execution of the loop is dominated by the execution of the greedy demand oracle that takes $O(n^2)$ time. This gives us a total running time of $O(n^2(\frac{M}{\delta}n + m))$. This produces a partition $S_1, \ldots, S_m$ such that $\sum_i v^i(S_i) \geq \sum_i v^i(S_i^*) - \delta n$ as we argued in Section 1. Taking $\delta = \frac{1}{2n}$, gives us: $\sum_i v^i(S_i) \geq \sum_i v^i(S_i^*) - \frac{1}{2}$ and therefore $\sum_i v^i(S_i) = \sum_i v^i(S_i^*)$ since both are integers. The running time in this case is $O(n^2(Mn^2 + m))$.

The previous version runs in pseudo-polynomial time due to the linear dependency on M. This can be easily improved to a dependency on $\log(M)$ by updating the prices in a multiplicative fashion. Initialize all prices to zero and define the following price update rule: $U(0) = \delta$ and $U(p_j) = p_j(1 + \delta)$. So each price is in the set $\{0, \delta, \delta(1+\delta), \delta(1+\delta)^2, \ldots\}$. Now, we change Algorithm 1 in two ways: first we calculate p^i as $p^i_j = p_j$ if $j \in S_i$ and $p^i_j = U(p_j)$ otherwise. Also, when we update

prices inside the while loop, we update p_j to $U(p_j)$. By the same argument as before, prices never rise past M, so there are at most $n \cdot \log_{1+\delta}(M)$ price updates. This produces a running time of $O(n^2 m + \frac{1}{\delta} n^3 \log M)$. The solution produced is such that $v^i(S_i) - p(S^i) \geq v^i(T) - p(T) - \delta|T \setminus S_i| - \delta p(T \setminus S_i)$ for all $T \subseteq [n]$. Taking $T = S_i^*$ (the optimal partition) and summing for all i, we get: $(1+\delta)\sum_i v^i(S_i) \geq \sum_i v^i(S_i^*) - \frac{n}{\delta}$.

Maximum matching. It is illuminating to look at the case of weighted maximum matching, i.e. $v^i(S) = \max_{j \in S} w_{ij}$. For this particular case, the Walrasian tatônnement procedure takes the form of the *auction method* from Bertsekas [7] and the ascending auction of Demange, Gale and Sotomayor [10]. It also closely resembles Kuhn's Hungarian Method [29]. For this particular case, the demand oracle can be computed in time $O(n)$, which gives us complexity $O(\frac{M}{\delta} n^2 + nm)$. Consider further the special case of unweighted maximum matching, where $n = m$, $w_{ij} \in \{0, 1\}$ and the graph has a perfect matching. Since $M = 1$, this gives an $(1-\delta)^{-1}$-approximation algorithm of running time $O(\frac{1}{\delta} n^2)$. Taking $\delta = \frac{1}{2}$ we get exactly the 2-approximation via the greedy algorithm for maximum matching. For $\delta = \frac{1}{n}$ we get an $O(n^3)$ exact algorithm. Taking $\delta = \frac{1}{\sqrt{n}}$ one gets a matching of size $n - \sqrt{n}$ in time, $O(n^2\sqrt{n})$. After more $\sqrt{n}$ iterations of an augmenting path algorithm, we are able to find the optimal matching with total running time $O(n^2\sqrt{n})$, which is the bound provided by the Hopcroft-Karp algorithm [25].

Gradient descent interpretation. The tatônnement procedure has been modernly reinterpreted as a form of gradient descent over a market potential [4, 43]. Ausubel [3] proposed the following potential function associated with market prices $p \in \mathbb{R}^n$:

$$\Phi(p) = \sum_{i=1}^{m} \max_{S_i \subseteq [n]} [v^i(S_i) - p(S_i)] + \sum_{j=1}^{n} p_j$$

We note that if $S_1^*, \ldots, S_m^*$ is the optimal allocation then:

$$\Phi(p) \geq \sum_i v^i(S_i^*) - p(S_i^*) + p([n]) = \sum_i v^i(S_i^*)$$

The inequality above holds with equality if and only if $S_i^* \in D(v^i; p)$ for each i. Therefore, a vector or prices is Walrasian if and only if it minimizes the potential Φ.

If for a certain price vector p, the demanded sets are $S_i \in D(v^i; p)$, then the vector $\mathbf{1}_{[n]} - \sum_i \mathbf{1}_{S_i}$ is a subgradient of Φ at p. Therefore, subgradient descent methods tend to increase the price of over-demanded goods, decrease the price of under-demanded goods, which is the guiding principle of the tatônnement procedure.

11.2. Linear Programming algorithms. The second approach, proposed by Nisan and Segal [40], is based on linear programming. They observe that the welfare problem can be cast as the following integer program:

$$
\begin{aligned}
W_{\text{IP}} = \max \sum_{i=1}^{m} \sum_{S \subseteq [n]} x_{iS} \cdot v^i(S) \qquad & \text{s.t.} \\
\sum_{S \ni j} \sum_{i} x_{iS} = 1, \quad & \forall j \in [n] \\
\sum_{S} x_{iS} = 1, \qquad & \forall i \in [m] \\
x_{iS} \in \{0,1\}, \qquad & \forall i \in [m], S \subseteq [n]
\end{aligned}
$$

Let W_{LP} correspond to the linear programming relaxation of the previous problem, i.e., to the program obtained by relaxing the last constraint to $0 \leq x_{iS} \leq 1$. Since it is a relaxation, $W_{\text{IP}} \leq W_{\text{LP}}$. Bikhchandani and Mamer [8] observe that when v^i are gross substitute valuations, this holds with equality, for the following reason: by the duality theorem in Linear Programming, W_{LP} corresponds to the solution of the following dual program:

$$
\begin{aligned}
W_{\text{LP}} = \min \sum_{i \in [m]} u_i + \sum_{j \in [n]} p_j \qquad & \text{s.t.} \\
u_i \geq v^i(S) - \sum_{j \in S} p_j, \quad & \forall i \in [m], S \subseteq [n] \\
p_j \geq 0, u_i \geq 0, \qquad & \forall i \in [m], j \in [n]
\end{aligned}
$$

An optimal solution to the integer program corresponds to the welfare of a Walrasian equilibrium $W_{\text{IP}} = \sum_i v^i(S_i)$. If p are the corresponding Walrasian prices and $u_i = v^i(S_i) - \sum_{j \in S_i} p_j$, (u, p) corresponds to a feasible solution to the dual. Therefore $W_{\text{LP}} \leq W_{\text{IP}}$.

Given this observation, Nisan and Segal propose solving the welfare problem by solving the dual linear program above using a separation based linear programming algorithm, such as the ellipsoid method. The program has $n + m$ variables but an exponential number of constraints. In order to solve it, we need to provide a separation oracle, i.e., an polynomial-time algorithm to decide, for each (u, p) if it is feasible and if not, produce a violated constraint. The problem that the separation oracle needs to solve is to decide for each agent i if $u_i \geq \max_S v^i(S) - \sum_{j \in S} p_j$. For gross substitute valuations, this can be easily solved by the greedy algorithm (Algorithm 2).

The algorithm described above computes the value $W^* = \sum_i v^i(S_i)$ of the optimal allocation. In order to compute the optimal allocation itself, one can proceed as follows: fix an arbitrary item j and for all agents i' compute the value of the optimal allocation, conditioned on agent i receiving j. In order words, compute the value of the optimal allocation of items $[n] \setminus j$ to agents with valuation $\tilde{v}^i = v^i$ for $i \neq i'$ and $\tilde{v}^{i'}(S) = v^{i'}(S|j)$ and let $W^*_{j \to i'}$ be the solution. Then there must be some agent i' for which $W^* = v^{i'}(j) + W^*_{j \to i'}$. Allocate j to this agent and recursively solve the allocation problem with on $[n] \setminus j$ for valuations $\tilde{v}$.

11.3. Flow-based algorithms. Finally we describe a purely combinatorial approach proposed by Murota [34, 35] based on a minimum cost flow formulation.

Murota's original algorithm is for a more general problem called the *assignment problem for valuated matroids*. In that problem there are sets two sets V_1 and V_2 together with matroids $\mathcal{M}_1 = (V_1, \mathcal{B}_1)$ and $\mathcal{M}_2 = (V_2, \mathcal{B}_2)$ defined on those sets. Given valuated matroids $\omega_1 : \mathcal{B}_1 \to \mathbb{R}$ and $\omega_2 : \mathcal{B}_2 \to \mathbb{R}$ and a weighted bipartite graph $G = (V_1, V_2, E, w)$, we are asked to compute a matching M maximizing the objective:

$$\max \omega_1(M_1) + \omega_2(M_2) + w(M) \text{ s.t. } M_1 \in \mathcal{B}_1, M_2 \in \mathcal{B}_2 \text{ and } M \text{ is a matching.}$$

where M_1 and M_2 are the endpoints of M in V_1 and V_2 respectively.

This problem is quite general and has as special cases several interesting combinatorial problems such as matroid intersection and the welfare problem for gross substitutes. Murota describes two strongly polynomial time algorithms for this problem: one based on cycle cancellations and one based on augmenting paths. Here will present the second one following the presentation in Paes Leme and Wong [32].

Murota's main idea to solve the welfare problem via flow augmentations is to incrementally compute the optimal allocation by successively solving the problem of finding the optimal allocation of the first t items (in some arbitrary order):

$$\max \sum_i v_i(S_i) \text{ s.t. } \cup_i S_i = [t] = \{1, \ldots, t\} \text{ and } S_i \cap S_j = \emptyset \text{ for } i \neq j \qquad (W_t)$$

The solution of (W_t) will consist of an allocation $S_1, \ldots, S_m$ of the first t items to the buyers together with a price vector $(p_1, \ldots, p_t)$ that certifies that the allocation is optimal, i.e., Walrasian prices for the economy with only t goods.

11.3.1. Algorithm description. We describe how to use a solution for problem (W_t) to obtain a solution for problem (W_{t+1}). We will start by building the exchange graph as specified in Definition 10.2 for the first $t+1$ items and using the allocation $S_1, \ldots, S_m$ obtained from (W_t). The graph is defined on $t+1+m$ nodes: the $t+1$ items and the m dummy items, which intuitively mean an empty spot in the allocation of each buyer. In order to simplify notation it will be convenient to extend the valuations v^i to sets that contain both items and dummy items by simply ignoring the dummy items in the valuations, i.e., for $S \subseteq [n+m]$, $v^i(S) = v^i(S \cap [n])$.

We will use the price potentials to modify the edges so that they will all become positive. Associate with each item $i \in [t]$ the price p_i obtained from (W_t) and associated with item $t+1$ a price p_{t+1} to be specified later and price $p_\phi = 0$ with each dummy item. We change the weight of edges from $j \in S'_i := S_i \cup \phi_i$ to $k \notin S'_i$

$$w_{jk} = -v^i(S_i \cup k \setminus j) + v^i(S_i) + p_k - p_j$$

From the optimality of the allocation for (W_t), all edges not involving item $t+1$ are non-negative, since no agent would prefer to add, drop or swap an item under the current prices. Since item $t+1$ is in none of the sets S_i, there are no edges going out of $t+1$. All the edges involving $t+1$ have weight with a positive $+p_{t+1}$ term. Such edges can be made positive by making p_{t+1} sufficiently high. The economic interpretation is that we start with the price of $t+1$ large enough that no other agent would like to buy it.

Allocation and price updates. Given the exchange graph constructed above, the algorithm to update the allocation is simple: we compute a shortest path with minimum length from a dummy item ϕ_i to item $t+1$. Since all the edges are non-negative

this can be done using Dijkstra's algorithm in $O((t+m)\log(t+m))$ time. Dijkstra's algorithm can be easily modified to compute the shortest path of minimum length. If the weights are integers, modify the weights w_{ij} by subtracting ϵ from edge for a small enough ϵ. Or more formally, we can run Dijkstra's algorithm on the ring $(\mathbb{Z}^2, +, <)$ where $+$ is the componentwise addition and $<$ is the lexicographic order.

Now the allocation can be updated by performing the changes prescribed by the path P in the output of Dijkstra's algorithm. An edge (j,k) with $j \in S_i$ means we swap j by k in S_i. An edge from (ϕ_i, k) means we add k to S_i. Formally if $(\{j_r, k_r)\}_{r=1..a}$ are the edges in P with $j_r \in S_i \cup \phi_i$, then construct $\tilde{S}_i = S_i \cup \{k_1, \ldots, k_a\} \setminus \{j_1, \ldots, j_r\}$.

Dijkstra's algorithm also produces a certificate of optimality which takes the form of a distance function for each node in the graph. The distance function has the property that $d(\phi_i) = 0$ and for any edge (j,k) we have:

$$d(k) \leq d(j) + w_{jk}$$

For edges $(i,j) \in P$, the previous equation holds with equality, i.e.,

$$d(k) = d(j) + w_{jk}$$

We can use the distance certificate to update the prices as follows:

$$\tilde{p}_j = p_j - d(j)$$

THEOREM 11.1 (Murota [35]). *If $S_1, \ldots, S_m$ is an optimal solution for problem (W_t) and $p_1, \ldots, p_t$ are optimal prices certifying this allocation, then the allocation $\tilde{S}_1, \ldots, \tilde{S}_m$ produced by the shortest path computation is an optimal solution to (W_{t+1}) certified by prices $\tilde{p}_1, \ldots, \tilde{p}_{t+1}$.*

If each set S_i has a single edge entering and a single edge leaving the set, then it is straightforward that the updated prices provide a certificate of optimality for the new allocation. In the general case, where there can be multiple edges entering and leaving the set, the proof will rely on two combinatorial lemmas which we state and prove in the next section.

11.3.2. Two combinatorial lemmas. The first lemma is a valuated version of the unique matching theorems for matroids:

LEMMA 11.2 (Unique Minimum Weight Matching Condition). *Given a gross substitute valuation v, a set S, $A = \{a_1, \ldots, a_k\} \subseteq S$, $B = \{b_1, \ldots, b_k\} \subseteq [n] \setminus S$, consider the bipartite graph G with left nodes A, right nodes B and edge weights $w_{a_i b_j} = v(S) - v(S \cup b_j \setminus a_i)$. If $M = \{(a_1, b_1), \ldots, (a_k, b_k)\}$ is the unique minimum weight matching in the graph, then:*

$$v(S) - v(S \cup B \setminus A) = \textstyle\sum_{j=1}^{k} v(S) - v(S \cup b_j \setminus a_j)$$

Proof. The proof of the lemma follows by induction on k. For $k = 1$, the theorem is trivial. Assume now it holds for $k-1$. First observe that: $v(S) - v(S \cup B \setminus A) = w_{a_k b_k} + v(S \cup b_k \setminus a_k) - v(S \cup B \setminus A)$. Define $\tilde{S} = S \cup b_k \setminus a_k$. If we can prove that the graph $\tilde{G}$ defined by left nodes $A \setminus a_k$, right nodes $B \setminus b_k$ and weights $\tilde{w}_{a_i b_j} = v(\tilde{S}) - v(\tilde{S} \cup b_j \setminus a_i)$ has $(a_1, b_1), \ldots, (a_{k-1}, b_{k-1})$ as the unique minimum weight matching and moreover $\tilde{w}_{a_i b_i} = w_{a_i b_i}$ then we can apply the induction hypothesis and conclude that:

$$v(S \cup b_k \setminus a_k) - v(S \cup B \setminus A) = \textstyle\sum_{i=1}^{k-1} \tilde{w}_{a_i b_i} = \sum_{i=1}^{k-1} w_{a_i b_i}$$

In order to $(a_1, b_1), \ldots, (a_{k-1}, b_{k-1})$ is the unique minimum matching, first we bound $\tilde{w}_{a_i b_j}$ and then we show that any other matching has strictly larger weight.

$$\begin{aligned}
\tilde{w}_{a_i b_j} &= v(S \cup b_k \setminus a_k) + v(S) - [v(S \cup \{b_k, b_j\} \setminus \{a_k, a_i\}) + v(S)] \\
&\overset{*}{\geq} v(S \cup b_k \setminus a_k) + v(S) - \\
&\qquad \max\{v(S \cup b_j \setminus a_i) + v(S \cup b_k \setminus a_k), v(S \cup b_j \setminus a_k) + v(S \cup b_k \setminus a_i)\} \\
&= \min\{w_{a_i b_j}, w_{a_i b_k} + w_{a_k b_j} - w_{a_k b_k}\}
\end{aligned}$$

where the $(*)$ inequality follows from Lemma 5.3.

Now, given a matching $\tilde{M}$ different then $(a_1, b_1), \ldots, (a_{k-1}, b_{k-1})$ on $\tilde{G}$, construct an auxiliary graph in which we add the following edges for each $(a_i, b_j) \in \tilde{M}$: (i) if $\tilde{w}_{a_i b_j} = w_{a_i b_j}$, then we add an edge between a_i and b_j with weight $w_{a_i b_j}$ and sign $+1$. (ii) if $\tilde{w}_{a_i b_j} = w_{a_i b_k} + w_{a_k b_j} - w_{a_k b_k}$ we add edges between a_i and b_k with weight $w_{a_i b_k}$ and sign $+1$, an edge between a_k and b_j with weight $w_{a_k b_j}$ and sign $+1$ and one edge between a_k and b_k with weight $w_{a_k b_k}$ and sign -1. By a simply counting argument, the *signed* degree of each node a_i or b_i with $i < k$ is 1 and the signed degree of nodes a_k, b_k is 0. Now we argue that the total signed weight of this graph is at least $\sum_{i=1}^{k-1} w_{a_i b_i}$. Indeed, if there are no edges incident to k this is obvious since M was the *unique* minimum matching in G. If there are edges incident to k, consider a cycle C containing edge (a_k, b_k) in the union between the M (with weight $w_{a_i b_i}$) and the $+1$-signed edges in the auxiliary graph. Let C_M be the edges in the cycle belonging to M and let $C_{\tilde{M}}$ be the remaining edges. Note that the total weight of C_M is strictly smaller then the total weight of $C_{\tilde{M}}$ since M is the unique minimum matching. Therefore, we remove the edges in $C_{\tilde{M}}$ from the auxiliary graph and add the edges in C_M, where adding an edge (a_k, b_k) with $+1$ sign is equivalent in removing one edge (a_k, b_k) with -1 sign. By repeating this procedure we eventually obtain an auxiliary graph with strictly smaller weight then the original and no incident edges on a_k, b_k. The weight of such graph must be at least $\sum_{i=1}^{k-1} w_{a_i b_i}$ since M is the unique minimum matching.

In order to finish the proof, we just need to argue that $\tilde{w}_{a_i b_i} = w_{a_i b_i}$. By the previous argument: $\tilde{w}_{a_i b_i} \geq \min\{w_{a_i b_i}, w_{a_k b_i} + w_{a_i b_k} - w_{a_k b_k}\} = w_{a_i b_i}$ since by the minimality of matching M, $w_{a_i b_i} + w_{a_k b_k} < w_{a_k b_i} + w_{a_i b_k}$. For the other direction, we again use Lemma 5.3 to see that:

$$\begin{aligned}
v(S \cup b_i \setminus a_i) + v(S \cup b_k \setminus a_k) \leq \max\{&v(S) + v(S \cup \{b_i, b_k\} \setminus \{a_i, a_k\}), \\
&v(S \cup b_k \setminus a_i) + v(S \cup b_i \setminus a_k)\} = v(S) + v(S \cup \{b_i, b_k\} \setminus \{a_i, a_k\})
\end{aligned}$$

since $w_{a_i b_i} + w_{a_k b_k} < w_{a_k b_i} + w_{a_i b_k}$ implies that $v(S \cup b_i \setminus a_i) + v(S \cup b_k \setminus a_k) > v(S \cup b_k \setminus a_i) + v(S \cup b_i \setminus a_k)$.
□

The next lemma shows that if we have a minimum length shortest path between two nodes in the exchange graph, then all edges leaving a set form an unique perfect matching like the statement of the previous lemma. Those two components together will allow us to overcome the main difficulty in proving Theorem 11.1.

LEMMA 11.3. *Let $G = (V, E, w)$ be a weighted directed graph with non-negative edges and P a shortest path from s to t. Given a set $S \subset V$, define a matching $M = \{(a, b) \in P; a \in S, b \notin S\}$ between the sets $A = \{a \in [n]; (a, b) \in M\}$ and $B = \{b \in [n]; (a, b) \in M\}$. If M is not the Unique Minimum Bipartite Matching*

in the bipartite graph with color classes A and B and weights corresponding to the weights of edges from A to B in G, then there is a shortest path from s to t with a strictly smaller number of edges.

Proof. Let $d(i)$ be the distance from the source s for each node and modify each weight to be $\tilde{w}_{ij} = w_{ij} + d(i) - d(j)$ such that each edge has $\tilde{w}_{ij} \geq 0$ and all the edges in the path P have $\tilde{w}_{ij} = 0$. Now, add to the graph an extra edge between t and s with $\tilde{w}_{ts} = 0$ and build a cycle C formed by P together with the newly added edge.

Assume now that there is an alternative perfect matching

$$M' = \{(a_{j_1}, b_{j_2}), (a_{j_2}, b_{j_3}), \ldots, (a_{j_k}, b_{j_1})\}$$

with total weight no larger then the original one. Consider now k cycles in graph G where the t-th cycle C_t is formed by edge $(a_{j_t}, b_{j_{t+1}})$ and the path from $b_{j_{t+1}}$ to a_{j_t} in the original cycle C. Each cycle C_t is either C (if this edge of M' is also in M) or is a cycle with smaller number of edges (otherwise).

Next we present a counting argument, there exists an integer ℓ such that the multi-set union of cycles $\{C_t; C_t \neq C\}$ has ℓ copies of each edge in $C \setminus M$, $\ell - 1$ copies of each edge in M and one copy of each edge in M'. The argument is as follows: transverse the cycle in the following way: start in b_{j_1} and follow cycle C until a_{j_k}, then take an edge (a_{j_k}, b_{j_k}) in M and transverse the cycle from b_{j_k} to $a_{j_{k-1}}$, take then the edge $(a_{j_{k-1}}, b_{j_{k-1}})$ in M and then transverse C from $b_{j_{k-1}}$ to $a_{j_{k-2}}$, continuing the same procedure until we take edge (a_1, b_1) in which case we complete an integral number ℓ of loops around the cycle. The transversal of from $b_{j^{t+1}}$ to a_{j^t} corresponds to the edges of cycle C_t minus the edge in M'. So if we look at the edges transverses add M' and subtract M, we get exactly the multi-set of edges in the union of the cycles C_t.

Therefore, The sum of modified weights $\tilde{w}_{ij}$ of such cycles is at most ℓ times the total modified weight in C which is zero. Since all edges are non-negative, then all cycles C_t must have total modified weight zero. One of them must contain the edge (t, s), so removing it, we obtain a path with modified weight zero from s to t that has less edges than P. Since the modified weight is zero, the total (unmodified) weight must be $d(t) - d(s)$ which is the total weight of P. □

11.3.3. Correctness and Running time analysis. The previous lemmas provide us the necessary tools to prove the correctness of the algorithm presented in Section 11.3.1.

Proof of Theorem 11.1. Let S_i be the optimal allocation for (W_t), p_j the prices certifying the optimality and $\tilde{S}_i$ and $\tilde{p}_j$ the new allocation and prices computed by the algorithm. Because $\tilde{p}_j$ is determined using the potentials output by Dijkstra's algorithm, if we redefine the weights as:

$$\tilde{w}_{jk} = -v^i(S_i \cup k \setminus j) + v^i(S_i) + \tilde{p}_k - \tilde{p}_j$$

for $j \in S'_i := S_i \cup \phi_i$ and $k \notin S'_i$, then we will still have $\tilde{w}_{jk} \geq 0$ for every edge and P will still be a shortest path with respect to the new weights. By Theorem 10.3, this means that the price vector $\tilde{p}$ also a valid certificate for (W_t), i.e.,

$$v^i(S_i) - \tilde{p}(S_i) \geq v^i(S') - \tilde{p}(S'), \quad \forall S' \subseteq [t+1]$$

Now, let $\{(j_r, k_r)\}_{r=1..a}$ be the set of edges in P such that $j_r \in S_i \cup \phi_i$. Lemma 11.3 guarantees that the this set of edges form an unique minimum bipartite matching, which is the condition required by Lemma 11.2 to guarantee that:

$$v^i(S_i) - v^i(\tilde{S}_i) = \sum_{r=1}^{a} v^i(S_i) + v^i(S_i \cup k_r \setminus j_r)$$

Summing $\tilde{p}(\tilde{S}_i) - \tilde{p}(S_i)$ on both sides, we get:

$$[v^i(S_i) - \tilde{p}(S_i)] - [v^i(\tilde{S}_i) - \tilde{p}(\tilde{S}_i)] = \sum_{r=1}^{a} \tilde{w}_{j_r k_r} = 0$$

Therefore:

$$v^i(\tilde{S}_i) - \tilde{p}(\tilde{S}_i) = v^i(S_i) - \tilde{p}(S_i) \geq v^i(S') - \tilde{p}(S')$$

which certifies the optimality of the $\tilde{S}_i$ allocation, completing the proof. Note that while we had a similar equation for S_i, the union $\cup_i S_i$ was missing item $t+1$. The update by the shortest path guaranteed that we still satisfy the condition that every buyer has his favorite bundle, while selling all the items. □

Running time analysis. Solving problem (W_t) involves building a graph with $t+m$ nodes and $O(mt + t^2)$ edges. Dijkstra's algorithm takes $\tilde{O}(mt + t^2)$ time giving an overall running time of $\tilde{O}(mn^2 + n^3)$. Paes Leme and Wong [32] observe that the total running time can be brought down to $\tilde{O}(mn + n^3)$ by collapsing all dummy nodes to one node ϕ with edges to each item j with weight

$$w_{\phi j} = p_j + \min_{i; j \notin S_i} [v^i(S_i) + v^i(S_i \cup j)]$$

Dijkstra's algorithm now takes $\tilde{O}(t^2)$ time per step, but we still need to compute the value of $w_{\phi j}$. Computing it naively gives us again extra $O(mn)$ extra computation per iteration. This can be improved by dividing the agents in active ($S_i \neq \emptyset$) and inactive ($S_i = \emptyset$). There are at most n active agents in any given time, so we can compute $\min_i [v^i(S_i) + v^i(S_i \cup j)]$ for each of them with total cost $O(n^2)$ per iteration. For the inactive agents we can pre-compute a data structure that keeps a sorted list L_j of $v^i(j)$ for each item j. Once an agent becomes active, we just remove it from all the lists (note once active, an agent stays active in every subsequent iteration). We can then query the minimum $\min_i [v^i(S_i) + v^i(S_i \cup j)]$ over inactive agents just by inspecting the smallest element in each list. The total running time becomes then $O(mn)$ pre-processing cost plus an additional $\tilde{O}(n^2)$ time per iteration which results in total running time of $\tilde{O}(mn + n^3)$.

11.3.4. Interpretation and Consequences. The algorithm can be seen as a descending price auction. The prices of the items start very high and one item at a time we decrease the price of one item while adjusting the prices of the others to keep an optimal allocation among the items considered so far.

The other interesting aspect of this algorithm is that it can be seen as an algorithm for dynamic markets with item insertion. After computing the equilibrium in a certain market, this algorithm allows to insert more items in the market and update the allocation by paying only the difference in computational cost.

Finally, the correctness proof of the algorithm implies some interesting structural properties of the optimal solution. The following corollary follows directly from the proof of Theorem 11.1:

COROLLARY 11.4. *Let $S_1, \ldots, S_m$ be the optimal allocation in a market with n items and m buyers with gross substitute valuations $v^1, \ldots, v^m$ and let $\tilde{S}_1, \ldots, \tilde{S}_m$ be the solution with one the items removed. The for all except one buyer: $|\tilde{S}_i| = |S_i|$ and for the remaining buyer $|\tilde{S}_i| = |S_i| - 1$.*

Proof. Simply note that the shortest path of minimum length 11.1 can't contain more than one dummy item, otherwise we can consider the path from the last dummy item to $t+1$ and obtain a path that is at least as short and has strictly less edges. Therefore all the changes prescribed by the path are swaps except for one, which is an insertion. □

12. Further remarks: Applications, Representation, Approximability and Relations to other classes. Gross substitutes have a wide range of applications in many different fields and therefore it is not surprising that this concept has been rediscovered many times in different contexts. In economics it was originally conceived as a natural condition to express substitutability among workers in two-sided labor markets. To the present date, the original paper by Kelso and Crawford [26] has more then 800 citations and the term "gross substitutes" appears in more then 3500 publications according in Google Scholar. A comprehensive exposition on all the work is out of the scope of this survey, but we point the reader to some key generalizations and applications of gross substitutes in economics that will serve as starting point to the reader: Roth [45], Hatfield and Milgrom [1], Sun and Yang [49, 50], Gul and Stachetti [21, 22] and Bikhchandani and Mamer [8].

In Discrete Mathematics, the concept of introduced by Dress and Wenzel [15, 14] to generalize the Grassmann-Plücker relations in p-adic analysis. Their prototypical example of a valuated matroid (which roughly corresponds to a gross substitute valuation defined only on sets of same cardinality, see Section 8 for a complete discussion) is the following function parametrized by a prime number p: $\omega_p : (\mathbb{Q}^n)^n \to \mathbb{Z}$ that associates every n-tuple of n-dimensional rational vectors $x_1, \ldots, x_n \in \mathbb{Q}^n$ to: $\omega_p(x_1, \ldots, x_n) = -\infty$ if $\det(x_1, \ldots x_n) = 0$ and $\omega_p(x_1, \ldots, x_n) = k$ if $\det(x_1, \ldots x_n) = p^{-k} \cdot \frac{a}{b}$ for integers a and b not divisible by p. The greedy algorithm for valuated matroids is presented in [15] as a generalization of the greedy algorithm that given a finite set of $\mathbb{Q}^n$-vectors spanning $\mathbb{Q}^n$, finds a basis minimizing the p-adic valuation of the determinant. We refer to Murota's book [36] for a comprehensive survey of applications of valuated matroids and gross substitute valuations to problems in engineering and operations research.

Surveys on gross substitutes. There are various great surveys focusing on different aspects of gross substitutability. Murota's monograph [37] is a great place to get a complete picture of the theory of discrete convex analysis. I also refer to different survey by Murota [38] on further applications to game theory and mechanism design, such as generalizations of the stable marriage problem. Shioura and Tamura [48] have an excellent survey on the multi-unit case, i.e., extending the theory presented here to functions with domain in the lattice $\mathbb{Z}^n$ instead of the hypercube. Another important extension of the concept of substitutability is from valuation functions to a more general class of objects called trading networks. For this view, I refer the reader

to the excellent text by Hatfield, Kominers, Nichifor, Ostrovsky and Westkamp [23].

12.1. Representation and Approximation. We finalize this survey with a discussion of the representation of gross substitute valuations and relations to other classes of valuation functions. We decided to keep the discussion of those points in the further remarks section since those are less well-understood properties, unlike the other topics presented in this survey. Also, unlike the rest of the survey, we assume for this section familiarity of the reader with Matroid Theory [30, 42].

One of the central issues in auction design is how to design a language in which agents can represent their valuation in a compact and expressible way. The fact that the demand oracle problem for gross substitute valuations allows for simple and efficient greedy algorithms might raise suspicion that valuations in this class may admit simple and compact representation. Also contributing to this suspicion is the observation by Lehmann, Lehmann and Nisan [31] that the set of gross substitute valuations has measure zero with respect to the set of all valuation functions. To make this statement precise, consider the representation of a valuation function $v : 2^{[n]} \to \mathbb{R}$ as a vector in $\mathbb{R}^{2^n-1}$ indexed by non-empty subsets of $[n]$. This allows for a geometric representation of a class of valuations functions $V \subseteq \{v : 2^{[n]} \to \mathbb{R}; v(\emptyset) = 0\}$ as a subset of the Euclidean space $\mathcal{G}(V) \subseteq \mathbb{R}^{2^n-1}$. The observation in [31] follows as a consequence of the isosceles triangle property of the ultrametric induced by gross substitute valuations (Iso) discussed in Section 4. Equation (Iso) implies that certain linear relations among the components of vector representation of each valuation function must be satisfied. Therefore $\mathcal{G}(\textsc{GrossSubstitutes})$ is contained in the union of finitely many hyperplanes and hence has measure zero. We also know from Section 2 that $\mathcal{G}(\textsc{GrossSubstitutes})$ is not convex. This is in sharp contrast to $\mathcal{G}(\textsc{Submodular})$ which is a full dimensional convex polyhedral set.

The observations in the previous paragraph motivate the question of whether gross substitute valuations admit a compact representation. To make this question precise, we need to define what we mean by *representation*. Intuitively, we will measure the size of a representation as the amount of memory a computer would require to store this valuation function. Also, to avoid discussing how to represent real numbers, we restrict our attention to integer-valued valuations, i.e., valuations of the type $v : 2^{[n]} \to \mathbb{Z}$. Each integer $z \in \mathbb{Z}$ requires $\lceil \log_2(z+1) \rceil$ of memory space in order to store its binary representation. The naive representation of any integer-valued valuation function such that $v(S) \leq M$ for each S requires at most $(2^n - 1) \cdot \lceil \log_2(M+1) \rceil$ space.

Certain classes of valuations can be represented with a lot less space, for example, unit demand valuations can be represented by n integers corresponding to the singleton values $v(\{i\})$ for all $i \in [n]$, therefore requiring $n \cdot \lceil \log_2(M+1) \rceil$ space. For any class of valuation we can ask the same question: what is the minimum amount of space necessary to represent an integer-valued valuations of this class over n items with values at most M ? The main reason to study this question is because often the representation size is a proxy for complexity and expressivity of a valuation function. This question is also closely related to counting the number of distinct valuation of this class over n items and value at most M. If $V(n, M)$ is this number, then it is possible to assign an unique index to each of those valuations and represent each such valuation by its index. Since each valuation is now represented by an index between 1 and $V(n, M)$, the size of representation is $\lceil \log_2 V(n, M) \rceil$. Although not a very useful representation, this gives a lower bound on the size of any given representation: any representation must have size at least $\lceil \log_2 V(n, M) \rceil$.

The exact space required to represent gross substitute functions is unknown. It

is strictly smaller then $(2^n - 1) \cdot \lceil \log_2(M+1) \rceil$ since by the discussion in Section 4, the values $v(S)$ satisfy various linear relations. Here we ask how smaller it is ? More precisely, can the exponential dependency on n be improved upon ?

A negative answer comes from the fact that matroid rank functions are a subclass of gross substitutes. A matroid rank function takes values in $\{0, 1, \ldots, n\}$ so admits a naive representation of size $O(2^n \log n)$. Knuth [27] proved that $\log_2 \log_2 m_n \geq n - O(\log n)$ where m_n is the total number of matroid rank functions with base set $[n]$. Since a representation of size s can encode at most 2^s different valuation functions, in order to represent all matroid rank functions we require at least $2^n/\text{poly}(n)$ size. This in particular implies that the exponential dependency in n can't be improved.

Balcan and Harvey [6] show that even if we settle for an approximation of the valuation function v, we still can't improve the exponential dependency on n. We say that $\hat{v}$ is an α-approximation of v if for every $S \subseteq [n]$, the relative error is at most α, i.e, $v(S) \leq \hat{v}(S) \leq \alpha \cdot v(S)$. They show in [6] that if $\alpha < \frac{n^{1/3}}{16 \log(n)}$, then an exponential dependency on n is still necessary for the size of the representation. Their is based on the following matroid construction:

THEOREM 12.1 (Balcan and Harvey [6]). *For every n and $\epsilon > 0$, there is a subset $\mathcal{X} \subseteq 2^{[n]}$ with $|\mathcal{X}| \geq 2^{n^{1/3-\epsilon}}$ such that for all $\mathcal{Y} \subseteq \mathcal{X}$, there exists a matroid rank function $r_{\mathcal{Y}} : 2^{[n]} \to \mathbb{R}$ such that $r_{\mathcal{Y}}(S) \geq n^{1/3}$ for all $S \in \mathcal{Y}$ and $r_{\mathcal{Y}}(S) \leq 8 \log(n)$ for all $S \in \mathcal{X} \setminus \mathcal{Y}$.*

If $\alpha < \frac{n^{1/3}}{16 \log(n)}$, any approximation $\hat{v}$ must be able to distinguish between the case $v(S) \geq n^{1/3}$ and $v(S) \leq 8 \log(n)$ for each $S \in \mathcal{X}$. The size of the representation must therefore be at least $2^{n^{1/3-\epsilon}}$.

The negative results on the representability of gross substitute functions arise from studying the subclass of matroid rank functions. A natural question that arises is: does the subclass of matroid rank functions encapsulate all difficulty in dealing with gross substitutes.

Ostrovsky and Paes Leme [41] propose to approach this question in the following way: defined the *weighted matroid* associated with matroid $\mathcal{M} \subseteq 2^{[n]}$ (represented my means of its independent sets) and weight vector $w \in \mathbb{R}^n_+$ as the function $v(S) = \max_{X \subseteq S, X \in \mathcal{M}} \sum_{j \in X} w_j$. Weighted matroids are gross substitute functions by Theorem 3.2. Now, define the class of *Matroid Based Valuations* as the smallest class that contains all weighted matroids and is closed under two operations: convolution (see Section 9) and endowment, which is the operation that given a valuation function $v : 2^{[n]} \to \mathbb{R}_+$ and a subset $S \subset [n]$, defines $w : 2^{[n] \setminus S} \to \mathbb{R}$ as $w(X) = v(X \cup S | S)$. Since the class of gross substitutes is closed under convolution and endowment, the class of matroid based valuations is a subclass of gross substitutes. It is an open question whether this inclusion is strict. The collapse of those two classes would offer an explanation of why the difficulty in dealing with gross substitute valuations seem to reside in the subclass of matroid rank function.

Another way to approach this question is via the notion of approximation: are there simple (or simpler) families of valuation functions that provide good approximations to gross substitute valuations ? Recall that a valuation function $\hat{v}$ is said to be an α-approximation for v if $v(S) \leq \hat{v}(S) \leq \alpha \cdot v(S)$ for all $S \subseteq [n]$. It is convenient at this point to restrict our attention to non-negative valued valuations, i.e., valuations in $V_+ = \{v : 2^{[n]} \to \mathbb{R}_+\}$. Given two classes of valuation functions $V_1, V_2 \subseteq V_+$, we

say that V_2 is α-approximated by V_1 if for all $v \in V_2$, there exists $\hat{v} \in V_1$ such that $\hat{v}$ is an α-approximation of v. Typically, $V_1 \subseteq V_2$ or at least, is in some sense 'simpler' then V_2. In that framework we can make the first question in this paragraph precise: what is the smallest α such that weighted matroids are an α-approximation to the class of gross substitute functions.

More broadly one could ask if there exist simple and natural classes of valuations that α-approximate the class of gross substitutes for small [5] values of α. The class of endowed assignment valuations proposed by Hatfield and Milgrom [1] (and discussed in Section 2) captures most practical examples of gross substitute valuations, but was recently shown by Paes Leme and Ostrovsky [41] to be a strict subclass. A natural question to ask if this class provides a good approximation to the class of gross substitutes.

One can also ask the same question in the opposite direction, how well do gross substitute valuations approximate more complex class of substitutes such as submodular and subadditive functions ? The approximability between other classes of valuations has been extensively studied [11, 20, 17, 5], yet very little is known about gross substitutes.

12.2. Algorithms for other classes of valuation functions. The various positive algorithmic results described in this survey provide encouragement to look at other valuation classes, in particular, wider valuation classes that also express the idea of subtitutability (such as submodular, subadditive and fractionally subadditive valuations [31]) and try to give algorithmic definitions for such classes, in the spirit of Theorems 3.2 and 3.4. The following paragraph hints that this is not be an easy task beyond gross substitutes.

For submodular valuations, it is known that both the greedy algorithm (Algorithm 2) and local search (Algorithm 3) can have arbitrarily bad performance for the demand oracle problem. Consider the following example: consider n items, among which n_r are red and n_b are blue. Now, given a set $S \subseteq [n]$, let S_r and S_b be the red and blue items in the set respectively. The following valuation function is submodular (but not gross substitutes):

$$v(S) = \min\{n_b \cdot |S_r| + n_r \cdot |S_b|, n_r \cdot n_b\}$$

Let $n_r \gg n_b$ and consider prices $p_i = 0$ for the red items and $p_i = n_r - n_b - \epsilon$ for the blue items. The greedy demand oracle chooses the set of all blue items which provide total utility of $n_b(n_b + \epsilon)$, while the demanded set under those prices is the set of all red items, which provides utility $n_b \cdot n_r$. For $\epsilon \to 0$, the ratio between the utility chosen by the greedy algorithm and then optimal utility is n_r/n_b which can be made as large as we want. The same example shows that local search can be equally as bad, since the set of all blue items is a local maximum with respect to the neighborhood defined in Algorithm 3. The situation is more severe: Feige, Immorlica, Mirrokni and Nazerzadeh [18] argue that no polynomial time algorithm can provide a constant factor approximation to the demand oracle problem for submodular functions (via a reduction to uniform set cover problem).

In the previous paragraph we argued that no polynomial time approximation algorithm is possible for the problem of finding a set of items maximizing $v(S) - p(S)$ for a generic submodular function v. Lehmann, Lehmann and Nisan [31] look at the

[5] Notice that this it not prevented by the result of Balcan and Harvey (Theorem **??**) since the simpler class of valuations might itself require the representation size to grow exponentially in n.

problem of finding a set S maximizing $v(S) + p([n] \setminus S)$ which is equivalent to the demand oracle problem from the perspective of exact algorithms. From the perspective of approximation, however, this is a completely different problem. They show that the greedy algorithm is a 2-approximation. In other words, they show that if S is the output of the greedy algorithm and S^* is a set in the demand set, then $v(S) + p([n] \setminus S) \geq \frac{1}{2}[v(S^*) + p([n] \setminus S^*)]$.

This suggests another direction in which one may hope to generalize Theorems 3.2 and 3.4 beyond gross substitutes: given an algorithm to compute the demand oracle, characterize the class of valuations for which those heuristics are exact or provide a good approximation to a suitable objective. For example, what is the class of valuations for which the procedure in [31] is a 2-approximation to the problem of maximizing $v(S) + p([n] \setminus S)$? Given a more powerful version of the local search procedure, what is the class of valuations for which it exactly computed demands ?

Acknowledgments. The author thanks Noam Nisan and Shahar Dobzinski for the initial discussions that led to this survey. He is also in debt with Moshe Babaioff and Oren Ben-Zwi for many detailed comments and suggestions to improve this manuscript. Thanks to Oren for providing a fix to the proof of Theorem 3.4 in an earlier version of this manuscript. The author also thanks Kazuo Murota, Akiyoshi Shioura, Sam Wong, Eric Balkanski, Sebastien Lahaie, Michael Ostrovsky, Brendan Lucier, Tim Roughgarden and Rakesh Vohra for suggestions and conversations which I clarified some of the points in this survey.

REFERENCES

[1] J. W. H. as matroid rank functions, gross substitutes. atfield, and P. R. Milgrom. Matching with Contracts. *American Economic Review*, 95(4):913–935, September 2005.

[2] L. Ausubel and P. Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1), 2002.

[3] L. M. Ausubel. An efficient dynamic auction for heterogeneous commodities. *The American economic review*, 96(3):602–629, 2006.

[4] L. M. Ausubel and P. R. Milgrom. Ascending auctions with package bidding. *Advances in Theoretical Economics*, 1(1), 2002.

[5] A. Badanidiyuru, S. Dobzinski, H. Fu, R. Kleinberg, N. Nisan, and T. Roughgarden. Sketching valuation functions. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1025–1035. SIAM, 2012.

[6] M.-F. Balcan and N. J. A. Harvey. Learning submodular functions. In *ECML/PKDD (2)*, pages 846–849, 2012.

[7] D. P. Bertsekas. The auction algorithm: a distributed relaxation method for the assignment problem. *Ann. Oper. Res.*, 14(1-4):105–123, June 1988.

[8] S. Bikhchandani and J. W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74(2):385–413, June 1997.

[9] M. Bing, D. J. Lehmann, and P. Milgrom. Presentation and structure of substitutes valuations. In *ACM Conference on Electronic Commerce*, pages 238–239, 2004.

[10] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–72, August 1986.

[11] S. Dobzinski. Two randomized mechanisms for combinatorial auctions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 89–103. Springer, 2007.

[12] A. Dress and W. Terhalle. Rewarding maps: On greedy optimization of set functions. *Advances in Applied Mathematics*, 16(4):464 – 483, 1995.

[13] A. Dress and W. Terhalle. Well-layered mapsa class of greedily optimizable set functions. *Applied Mathematics Letters*, 8(5):77 – 80, 1995.

[14] A. W. Dress and W. Wenzel. Valuated matroids: a new look at the greedy algorithm. *Applied Mathematics Letters*, 3(2):33 – 35, 1990.

[15] A. W. Dress and W. Wenzel. Valuated matroids. *Advances in Mathematics*, 93(2):214–250, 1992.
[16] S. Dughmi, T. Roughgarden, and Q. Yan. Optimal mechanisms for combinatorial auctions and combinatorial public projects via convex rounding. *Journal of the ACM (JACM)*, 63(4):30, 2016.
[17] U. Feige. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing*, 39(1):122–142, 2009.
[18] U. Feige, N. Immorlica, V. S. Mirrokni, and H. Nazerzadeh. Pass approximation: A framework for analyzing and designing heuristics. *Algorithmica*, 66(2):450–478, 2013.
[19] S. Fujishige and Z. Yang. A note on kelso and crawford's gross substitutes condition. *Math. Oper. Res.*, 28(3):463–469, July 2003.
[20] M. X. Goemans, N. J. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 535–544. Society for Industrial and Applied Mathematics, 2009.
[21] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, July 1999.
[22] F. Gul and E. Stacchetti. The english auction with differentiated commodities. *Journal of Economic Theory*, 92(1):66–95, May 2000.
[23] J. W. Hatfield, S. D. Kominers, A. Nichifor, M. Ostrovsky, and A. Westkamp. Full substitutability. In *Proc. of the 16th ACM Conference on Electronic Commerce (EC'15). ACM*, 2015.
[24] H. Hirai and K. Murota. M-convex functions and tree metrics. *Japan Journal of Industrial and Applied Mathematics*, 21(3):391–403, 2004.
[25] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.
[26] J. Kelso, Alexander S and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50(6):1483–1504, November 1982.
[27] D. E. Knuth. The asymptotic number of geometries. *Journal of Combinatorial Theory, Series A*, 16(3):398 – 400, 1974.
[28] B. Korte, L. Lovász, and R. Schrader. *Greedoids*. Algorithms and Combinatorics. Springer-Verlag, 1991.
[29] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955.
[30] E. Lawler. *Combinatorial optimization: networks and matroids*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 1976.
[31] B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.
[32] R. P. Leme and S. C.-w. Wong. Computing walrasian equilibria: fast algorithms and structural properties. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 632–651. Society for Industrial and Applied Mathematics, 2017.
[33] K. Murota. Convexity and steinitz's exchange property. *Advances in Mathematics*, 124(2):272 – 311, 1996.
[34] K. Murota. Valuated matroid intersection i: Optimality criteria. *SIAM J. Discrete Math.*, 9(4):545–561, 1996.
[35] K. Murota. Valuated matroid intersection ii: Algorithms. *SIAM J. Discrete Math.*, 9(4):562–576, 1996.
[36] K. Murota. *Matrices and matroids for systems analysis*, volume 20. Springer, 2000.
[37] K. Murota. *Discrete convex analysis*. SIAM, 2003.
[38] K. Murota. Discrete convex analysis: A tool for economics and game theory. *Journal of Mechanism and Institution Design*, 1(1):151–273, 2016.
[39] K. Murota and A. Shioura. M-convex function on generalized polymatroid. *Mathematics of Operations Research*, 24(1):pp. 95–105, 1999.
[40] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *J. Economic Theory*, 129(1):192–224, 2006.
[41] M. Ostrovsky and R. P. Leme. Gross substitutes and endowed assignment valuations. *Theoretical Economics*, 2014.
[42] J. Oxley. *Matroid theory*. Oxford Graduate Texts in Mathematics Series. Oxford University Press, Incorporated, 1992.
[43] D. C. Parkes and L. H. Ungar. An ascending-price generalized vickrey auction. 2002.
[44] H. Reijnierse, A. v. Gellekom, and J. A. M. Potters. Verifying gross substitutability. *Economic Theory*, 20(4):pp. 767–776, 2002.
[45] A. E. Roth. Stability and polarization of interests in job matching. *Econometrica*, 52(1):47–57,

1984.

[46] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Number v. 1 in Algorithms and Combinatorics. Springer, 2003.

[47] L. Shapley. *Complements and Substitutes in the Optimal Assignment Problem*. ASTIA document. Rand Corporation, 1958.

[48] A. Shioura and A. Tamura. Gross substitutes condition and discrete concavity for multi-unit valuations: a survey. *Journal of the Operations Research Society of Japan*, 58(1):61–103, 2015.

[49] N. Sun and Z. Yang. Equilibria and indivisibilities: gross substitutes and complements. *Econometrica*, 74(5):1385–1402, 2006.

[50] N. Sun and Z. Yang. A double-track adjustment process for discrete markets with substitutes and complements. *Econometrica*, 77(3):933–952, 2009.

[51] L. Walras. *Elements of Pure Economics: Or the Theory of Social Wealth*. Elements of Pure Economics, Or the Theory of Social Wealth. Taylor & Francis, 2003.