

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 tâtonnement* – 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 substitutability 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 convex concave: 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 tâtonnement procedure, linear programming approaches and purely combinatorial strongly-polynomial time algorithms.

1. Gross substitutes and Walrasian tâtonnement. The notion of *gross substitutes* was introduced by Kelso and Crawford [14] 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 work is over-demanded. Gul and Stacchetti [11] 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 $[n] = \{1, \dots, n\}$ a set of items (goods). A *valuation* over such items is a function $v : 2^{[n]} \rightarrow \mathbb{R}$ such that $v(\emptyset) = 0$. Given a price vector $p \in \mathbb{R}^n$, 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 $v(\{i, j\}|S)$.

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 $v^i : 2^{[n]} \rightarrow \mathbb{R}$. We use the notion of the *demand correspondence* to define an equilibrium of this economy:

DEFINITION 1.1 (demand correspondence). *Given a valuation function $v : 2^{[n]} \rightarrow \mathbb{R}$ and a vector of prices $p \in \mathbb{R}^n$, we define the demand correspondence as:*

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

DEFINITION 1.2 (Walrasian equilibrium). *Given an economy with indivisible goods with n goods, m agents and valuations v^i , a Walrasian equilibrium corresponds to a vector of prices $p \in \mathbb{R}_+^n$ and a partition of the goods in disjoint sets $[n] = \cup_{i=1}^m S_i$*

*Microsoft Research Silicon Valley, (renatop@microsoft.com).

such that $S_i \in D(v^i, p)$.

A reader familiar with the duality theorem in linear programming will readily recognize that the definition of Walrasian equilibrium closely resemble *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, \dots, 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^*, \dots, 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, \dots, S_m) is a Walrasian equilibrium and S_1^*, \dots, S_m^* maximizes $\sum_i v_i(S_i^*)$, then (p, S_1^*, \dots, 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 condition that allows such process to converge. The general idea behind this procedure goes back to Walras' *tatônnement procedure* [27], 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 (overdemanded) we increase the price. For each item that is demanded by no agent (underdemanded), we decrease the price. We iterate this until no item is overdemanded or underdemanded.

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 .

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. $p_j^i = p_j$ if $j \in S_i$ and $p_j^i = p_j + \delta$ o.w.

while there exists i such that $S_i \notin D(v^i, p^i)$

find a demanded set under the this 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 $j \neq i$

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: $v^i(S_i) - p(S_i) \geq v^i(T) - p(T) - \delta|T \setminus S_i|$.

In the limit as $\delta \rightarrow 0$, 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 $(p^t, S_1^t, \dots, S_m^t)$ be the outcome of the Walrasian tatonnement procedure for $\delta_t = \frac{1}{t}$ for $t \in \mathbb{Z}_+$. Since there are finitely many allocations (S_1^t, \dots, S_m^t) , there is one allocation that happens infinitely often. Let S_1, \dots, S_m be such allocation and let $t_1 < t_2 < \dots$ be the infinite subsequence corresponding to this allocation. Since p^t is bounded, passing to a subsequence if necessary, we can assume that $p^t \rightarrow p$. So taking $t \rightarrow \infty$ 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 tatonnement 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 during the execution of the algorithm, no item is even un-allocated. This motivates the following definition:

DEFINITION 1.3 (gross substitutes, Kelso and Crawford [14]). *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 demands the items in S whose price didn't increase.

THEOREM 1.4 (Kelso and Crawford [14]). *If valuations v^1, \dots, 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 [11] 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 $v^1, \dots, v^m \in \mathcal{C}$ there is a Walrasian equilibrium, then all valuations in \mathcal{C} are gross substitutes.

2. Examples, Non-Examples, OXS 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. The most natural subclass of gross substitutes is the class of matroid rank functions (we refer to Lawler [17] or Oxley [25] for a comprehensive discussion on matroids). Yet, sum of matroid rank functions might *not* be gross substitute valuations. For example, given three items $\{a, b, c\}$ define the function $r^i : 2^{\{a, b, c\}} \rightarrow \mathbb{R}$ 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 any $i = a, b, c$, r^i is a matroid rank function and hence satisfy the gross substitutability. However, the val-

uation $v = r^a + 2 \cdot r^b + 3 \cdot r^c$ 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 c makes b not be in any demanded set, violating Definition 1.3.

One other traditional example of gross substitutes are (i) *unit-demand functions*, such that $v(S) = \max_{i \in S} v(\{i\})$, (ii) *additive functions*, which are functions such that $v(S) = \sum_{i \in S} v(\{i\})$ and (iii) *symmetric submodular functions*, which are of the form $v(S) = f(|S|)$ for some monotone concave function $f : \mathbb{R}_+ \rightarrow \mathbb{R}_+$. The previous three examples are special cases of the class of OXS functions, introduced by Lehmann, Lehmann and Nisan [18].

DEFINITION 2.1 (OXS [18]). *A valuation function $v : 2^{[n]} \rightarrow \mathbb{R}_+$ is in the OXS class if there is a bipartite graph with non-negative weights on the edges, and left vertex-set corresponding to $[n]$ such that $v(S)$ is the weight of the maximum weighted matching on the subgraph induced by the right side nodes and the set S on the left side.*

We postpone until Section 8 a proof that all OXS valuations are gross substitutes, but we remark that unit-demand functions are the special case where there is only one vertex on the right side, additive functions correspond to the case where the vertices on the right side have degree 1 and the symmetric submodular case corresponds to the case where the bipartite graph is complete and all edges incident to a right-side node have the same weight.

Gul and Stachetti [11] observe that gross substitutes are a subclass of *submodular functions*:

DEFINITION 2.2 (submodularity). *A valuation function is said to be submodular 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 2.3 (Gul and Stachetti [11]). *Every gross substitute valuation function is submodular.*

Proof. Let v be a gross substitute valuation functions. Given $S \subseteq [n]$ and $i, j \notin S$, consider the price vector¹ such that $p_t = \infty$ for $t \notin S \cup \{i, j\}$, $p_t = -\infty$ for $t \in S \cup \{j\}$ and $p_i = v(i|S \cup \{j\})$. Clearly $S \cup \{i, j\} \in D(v, p)$. 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: $v(i|S) \geq v(i|S \cup \{j\})$. \square

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, \dots, w_n . The following example

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

due to Lehmann, Lehmann and Nisan [18] 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 $p = [1/2, 1/2, 1]$, $D(v, p) = \{\{a, b\}, \{c\}\}$, but if the price of a increases and the price vector becomes $p' = [1, 1/2, 1]$, 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 algorithmic primitive needed to implement the Walrasian tâtonnement 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:

Algorithm 2 Greedy demand oracle

Input: $p \in \mathbb{R}_+^n$, $v : 2^{[n]} \rightarrow \mathbb{R}_+$

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

The demand oracle problem can be used to give an alternative definition of gross substitutes as the class of valuations for which the greedy heuristic is exact:

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

The first part of this survey will be devoted to connect the classical definition 1.3 of gross substitutes to the definition 3.1. The path connecting those two definitions involves insights by different authors and reveals, in the route, many interesting properties about gross substitutes.

Before we proceed with this task, 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 or removing one element.

Algorithm 3 Local search demand oracle

Input: $p \in \mathbb{R}_+^n$, $v : 2^{[n]} \rightarrow \mathbb{R}_+$ Let $X = X_0 \subseteq [n]$ be an arbitrary initial set

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 [11] 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 3.2 (gross substitutes, Gul and Stacchetti [11]). *A valuation function satisfies the gross substitutes property if for all price vectors $p \in \mathbb{R}^n$, the local search algorithm implements a demand oracle, i.e., $L(v, p, X_0) \in D(v, p)$, where $L(v, p, X_0)$ is the output by the local search demand oracle (Algorithm 3).*

Equivalently, a valuation function satisfies the gross substitutes property 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 $j \in S$.

4. A price independent local definition. We make a brief detour and look at a different, yet very related question about gross substitutes. All definitions 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 definition of gross substitutes was resolved simultaneously by Fujishige and Yang [10] and Reijnierse, Gellekom and Potters [26]. The first paper provides a powerful connection to the theory of Discrete Convex Analysis, which we discuss in more detail in Section 7. We focus first on the definition given by Reijnierse et al [26].

THEOREM 4.1 (Reijnierse, Gellekom and Potters [26]). *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:*

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

The high level picture of their proof is quite simple and illuminating. Here we provide a brief sketch of it. First, they show that if a function doesn't have the gross substitutes property iff it is possible to find the following certificate: a price vector $p \in \mathbb{R}^n$ such that

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

Notice that the existence of such certificate clearly shows the violation of gross substitutability: the increase in the price of j would make the demand set become $\{S\}$ in case (i) and $\{S \cup \{k\}\}$ in case (ii). Therefore, item i would no longer be in the demand set. Proving the other direction requires more work, but its essence is to search for a minimal violation of gross substitutability by starting with an arbitrary one and changing the price vector so to shrink the demand set until it is minimal.

The second step is to transform the existence of certificates as above 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 (GS).

First, assume that we have 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).

Assume now that we have 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 (GS) 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).

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 $v_{R|S} : 2^R \rightarrow \mathbb{R}$ 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 4.2. *A valuation function satisfies the gross substitutes property iff every 3-restriction satisfied the gross substitutes property.*

An equivalent characterization of the one in Theorem 4.1 was also observed in Lehmann, Lehmann and Nisan [18] and Bing, Lehmann and Milgrom [4]. The latter characterization defines for each valuation v and subset $S \subseteq [n]$ a measure of how two goods i and j are substitutes to each other. Given $i, j \notin S$, let:

$$\alpha_S(i, j) = v(i|S) + v(j|S) - v(i, j|S)$$

and observe that (GS) is equivalent to $\alpha_S(i, j) \geq \min[\alpha_S(i, k), \alpha_S(k, j)]$. This in particular says that $d_S(i, j) = \alpha_S(i, j)^{-1}$ is a metric satisfying the following stronger version of the triangle inequality: $d_S(i, j) \leq \max[d_S(i, k), d_S(k, j)]$. Such metrics are called *ultrametrics* and have the interesting properties that all triangles are isosceles. This translates back to α_S as saying that given $\{i, j, k\}$, then up to renaming we have:

$$\alpha_S(i, k) = \alpha_S(k, j) \leq \alpha_S(i, j) \quad (\text{Iso})$$

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

LEMMA 4.3. *Given a gross substitute valuation $v : 2^{[n]} \rightarrow \mathbb{R}$, $S \subseteq [n]$ and $i_1, i_2, j_1, j_2 \notin S$, then:*

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

Let $P = \{i_1, i_2, j_1, j_2\}$ and $M = \min_{\{t_1, t_2\} \in P} \alpha_S(t_1, t_2)$. Define the *length* of the edge between (t_1, t_2) as $\alpha_S(t_1, t_2)$. By property (Iso), every triangle is isosceles with the smaller edge appearing at least twice. So, if we look the graph of the edges of minimal length between P , then either one of two things happen: (i) there is a cycle, i.e., there are $\alpha_S(x_1, x_2) = \alpha_S(x_2, x_3) = \alpha_S(x_3, x_4) = \alpha_S(x_4, x_1) = M$ where $\{x_1, x_2, x_3, x_4\} = \{i_1, i_2, j_1, j_2\}$ or (ii) there is a star, i.e., $\alpha_S(x_1, x_2) = \alpha_S(x_2, x_3) = \alpha_S(x_2, x_4) = M$. In order to see that, let x_1, x_2 be nodes in P such that $\alpha_S(x_1, x_2) = M$. Let x_3 be some other node, so the triangle x_1, x_2, x_3 must have two sides of length M . Up to renaming, $\alpha_S(x_1, x_2) = \alpha_S(x_2, x_3) = M$. If $\alpha_S(x_2, x_4) = M$ we are in case

(ii). If $\alpha_S(x_2, x_4) > M$ then by property (Iso) applied on the triangles x_1, x_2, x_4 and x_2, x_3, x_4 , we must have $\alpha_S(x_1, x_4) = \alpha_S(x_3, x_4) = M$, in which case we are in case (i).

Now, proving the statement of the lemma in each of the two cases is simple: if we are in case (i), then we can assume (swapping the names of i_1, i_2 if necessary) that $\alpha_S(i_1, j_1) = \alpha_S(i_2, j_2) = M$, so: $\alpha_S(i_1, j_1) + \alpha_S(i_2, j_2) = 2M \leq \alpha_S(i_1, i_2) + \alpha_S(j_1, j_2)$, which is equivalent to the statement in the lemma. In case (ii), say i_1 is the center of the star, i.e., $\alpha_S(i_1, x) = M$ for all $x \in \{i_2, j_1, j_2\}$. Now: $\alpha_S(j_1, j_2) \geq \min[\alpha_S(i_2, j_1), \alpha_S(i_2, j_2)]$. Swapping the names of j_1 and j_2 if necessary, we can assume that: $\alpha_S(j_1, j_2) \geq \alpha_S(i_2, j_1)$. That together with $\alpha_S(i_1, i_2) = M = \alpha_S(i_1, j_2)$, we get again $\alpha_S(i_1, j_1) + \alpha_S(i_2, j_2) \leq \alpha_S(i_1, i_2) + \alpha_S(j_1, j_2)$ which is equivalent to the statement in the lemma.

5. Well Layered and Matroidal Maps. The final step towards proving that the definitions 3.1 and 1.3 is the concept of *well layered maps* introduced by Dress and Terhalle [7] – in which the authors characterize the set functions $v : 2^{[n]} \rightarrow \mathbb{R}$ for which greedy algorithms are optimal.

DEFINITION 5.1 (well-layered map). *A function $v : 2^{[n]} \rightarrow \mathbb{R}$ is called well-layered iff for each $p \in \mathbb{R}^n$ the sets S_0, S_1, S_2, \dots obtained by the greedy algorithm (i.e., $S_0 = \emptyset$ and $S_i = S_{i-1} \cup \{x_i\}$ where $x_i \in \operatorname{argmax}_{x \in [n] \setminus S_{i-1}} v_p(x|S_{i-1})$) are such that $v_p(S_i) = \max\{v_p(S); |S| = i\}$.*

THEOREM 5.2 (Dress and Terhalle [7]). *A map $v : 2^{[n]} \rightarrow \mathbb{R}$ is well-layered iff for any triple of disjoint sets $S, \{i\}, T$ with $|T| \geq 2$,*

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

One readily recognizes condition (GS) to be a special case of (WL) with $|T| = 2$. In fact, they are equivalent. Let's show that (GS) implies (WL) by induction on $|T|$. 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 $|T| = t$.

Now, we give a sketch of the proof for Theorem 5.2. 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 $\{x_1, \dots, x_k\} \subseteq S'$. Since $|S'| = |S_t| = t$, clearly $k < t$. Now, applying (WL) for

$\{x_1, \dots, x_k\}, x_{k+1}, T' = S' \setminus \{x_1, \dots, x_k\}$ we get that there is $j \in T'$ such that:

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

and since $v(x_{k+1}|x_1, \dots, x_k) \geq v(j|x_1, \dots, x_k)$ 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 (GS), (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 5.2.

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 [6] 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 5.3 (Dress and Terhalle [6]). *A valuation function satisfied Definition 3.1 iff it is well-layered and submodular.*

The proof is simple: notice that submodularity guarantees that for 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 $v(i, j|S) > v(i|S) + v(j|S)$. 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 $S \cup \{i, j\}$.

Lemma 5.3 establishes our main claim that Definitions 3.1 and 1.3 are equivalent. We would like to finish by pointing out that many combinatorial structures such as matroids, polymatroids, valuated matroids [8], 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* [15]. Similar arguments can be used to argue about local search:

THEOREM 5.4. *A valuation function satisfied Definition 3.2 iff it is well-layered and submodular.*

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 $S^* \in \operatorname{argmax}_{S \subseteq [n]} v_p(S)$

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) $S^* \subseteq S$ For this case, $0 > v_p(S \setminus S^* | S^*) \geq v_p(i | S \setminus i)$. So there $i \in S \setminus S^*$ 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 $v_p(i | S \cap S^*)$. 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:

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

Since $v_p(i | S \cap S^*) \leq v_p((S \setminus S^*) \cup i \setminus j | S \cap S^*)$ we have that $v_p(S) \leq v_p(S \cup i \setminus j)$. If on the other hand, $i \in S$, we use 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 to show that in such case the greedy algorithm can be suboptimal.

6. Duality theorem for gross substitutes. The duality between gross substitutes and submodular functions was observed in many places, as in Fujishige and Yang [10] and Murota [20], Gul and Stacchetti [12] and Ausubel and Milgrom [1]. Given a valuation function v , they consider the utility function $u : \mathbb{R}^n \rightarrow \mathbb{R}$ which maps a set of prices $p \in \mathbb{R}^n$ to the optimal utility that can be obtained under such prices $u(p) = \max_S v_p(S)$. They relate it to the concept of \mathbb{R}^n -valued submodular function, which are functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 6.1 (duality, Ausubel and Milgrom [1]). *A valuation function $v : 2^{[n]} \rightarrow \mathbb{R}$ has the gross substitutes property iff its associate utility $u(p) = \max_S v_p(S)$ is an \mathbb{R}^n -valued submodular function.*

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 :

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

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 6.2 (duality, Gul and Stacchetti [12]). *A valuation function $v : 2^{[n]} \rightarrow \mathbb{R}$ 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 5.4.

7. Connection to Discrete Convex Analysis and Valuated Matroids.

Fujishige and Yang [10] 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 [20] that defines a very general class of functions $f : \mathbb{Z}^n \rightarrow \mathbb{R}$ 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 [23] define M^{\natural} -concave functions based on the concept of M -concavity of Murota [20]. 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 7.1 (M^{\natural} -concave functions, Murota and Shioura [23]). *A function $v : 2^{[n]} \rightarrow \mathbb{R}$ is M^{\natural} -concave if for all $S, T \subseteq [n]$ and $s \in S \setminus T$,*

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

THEOREM 7.2 (Fujishige and Yang [10]). *A function $v : 2^{[n]} \rightarrow \mathbb{R}$ has the gross substitutes property iff it is M^{\natural} -concave.*

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

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

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 is trivial. For $|S \setminus T| = |T \setminus S| = 2$, this follows directly from Lemma 4.3. 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).

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 [8]:

DEFINITION 7.3. Let $\binom{[n]}{k} = \{S \subseteq [n]; |S| = k\}$. We say that a map $\omega : \binom{[n]}{k} \rightarrow \mathbb{R}$ is a valuated matroid if it is satisfied 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 7.2 together with the discussion in its proof imply that:

LEMMA 7.4. A valuation $v : 2^{[n]} \rightarrow \mathbb{R}$ satisfies the gross substitutes property iff the map $\omega : \binom{[2n]}{n} \rightarrow \mathbb{R}$ 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.

8. 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 8.1 (Lehmann, Lehmann and Nisan [18] and Murota [20]). *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:*

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

We will give an algorithmic proof of the previous theorem based on a couple of observations about the Walrasian tâtonnement 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 tâtonnement 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 tâtonnement 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, \dots, 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 8.1. 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 tâtonnement 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 tâtonnement 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, $(S_1 \cup S_2) \cap \{j; p_j = p'_j\} \subseteq S'_1 \cup S'_2$.

A corollary of Theorem 8.1 is that OXS valuations are gross substitutes: it is straightforward from the definition that an OXS valuation function can be written as a convolution of unit-demand functions, one for each right-side node in the bipartite graph.

9. 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, \dots, v^m has two components: the first is called the *welfare problem*, which consists in finding a partition S_1, \dots, 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 tâtonnement procedure. Nisan and Segal [24] propose a solution that explores properties of gross substitutes to build a suitable linear program. Finally, Murota [21, 22] 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 [11]:

LEMMA 9.1 (Gul and Stachetti [11]). *Let W be the optimal welfare of an economy with a set $[n]$ of items and agents with gross substitute valuations v^1, \dots, 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 [21]:

LEMMA 9.2 (Murota [21]). *Let S_1, \dots, S_m be the optimal allocation of a set $[n]$ of items to agents with gross substitute valuations v^1, \dots, v^m . Add m dummy items to the set, i.e., extend the original set of items to $[n+m] = [n] \cup \{d_1, \dots, d_m\}$ in such a way that for each set S , $v^i(S) = v^i(S \cap [n])$. Now, define $S'_i = S_i \cup d_i$ and a graph G with nodes $[n+m]$ and weighted directed edges*

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

If the allocation is optimal, the graph has no negative cycles and therefore, the shortest-path distance is well defined. For each $i \in [n+m]$, let ϕ_i be the distance from dummy node d_1 to i . Then $\phi_i \leq 0$ and the vector $p_i = -\phi_i$ is a vector of Walrasian prices.

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 d_i, d_j , the weight of the arcs $w_{d_i, d_j} = w_{d_j, d_i} = 0$, then $\phi_{d_i} = 0$ for all dummy nodes. Also, for all items $k \in S_i$, the weight of the arc from a dummy node d_i to k is $w_{d_i, k} = -v^i(S_i \cup k) + v^i(S_i) \leq 0$, so $\phi_k \leq 0$ for all nodes k . Finally, notice that since ϕ is the shortest path distance, for all $j \in S'_i, k \notin S'_i$: $\phi_k \leq \phi_j + w_{jk}$, 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 5.4. 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 $\phi_j = -p_j$ for all $j \in [n]$ and $\phi_d = 0$ for all dummy items d . By the same argument as above, the fact that p is a vector of Walrasian prices implies that: $w_{ij} \geq \phi_i - \phi_j$, therefore for every cycle $i_1, i_2, \dots, i_k, i_1$ we have: $\sum_{t=1}^k w_{i_t i_{t+1}} \geq \sum_{t=1}^j \phi_{i_t} - \phi_{i_{t+1}} = 0$.

10. 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 where the items correspond to edges of the graph, there are k agents 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.

10.1. 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 $1, \dots, m$. 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, \dots, 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 $\{0, \delta, \delta(1 + \delta), \delta(1 + \delta)^2, \dots\}$. Now, we change Algorithm 1 in two ways: first we calculate p^i as $p_j^i = p_j$ if $j \in S_i$ and $p_j^i = U(p_j)$ 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 $O(n^2m + \frac{1}{\delta}n^3 \log M)$. 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 [2] and the ascending auction of Demange, Gale and Sotomayor [5]. It also closely resembles Kuhn's Hungarian Method [16]. 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 [13].

10.2. Linear Programming algorithms. The second approach, proposed by Nisan and Segal [24], is based on linear programming. They observe that the welfare problem can be cast as the following integer program:

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

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 [3] 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:

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

Given an optimal solution to the integer programming 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).

10.3. Cycle-cancelling algorithms. Finally we describe a purely combinatorial approach proposed by Murota [21, 22] based on the Fujishige's cycle-cancelling technique [9]. This approach has the advantage that it leads to a strongly polynomial time algorithm.

Murota's optimality criteria (Lemma 9.2) states that if an allocation S_1, \dots, S_m is not optimal, the directed graph as described in Lemma 9.2 has a negative cycle. The graph will contain a negative-weight cycle iff the allocation induced by S_1, \dots, S_m is not efficient.

Let C be this negative weight cycle. First we note that an edge going out of S'_i corresponds to the exchange of a (possibly dummy) element $a^i \in S'_i$ by an element $b^i \notin S'_i$. The value of the edge corresponds to the change in value for i by replacing a^i by b^i , i.e., $w_{a^i b^i} = v^i(S_i) - v^i(S_i \cup b^i \setminus a^i)$.

This observation suggests the following approach to improve the allocation: perform the exchanges prescribed by such cycle, i.e., if $M^i = \{(a_1^i, b_1^i), \dots, (a_{k^i}^i, b_{k^i}^i)\}$ is the set of edges in C going from S'_i , then update S_i to $S_i \cup \{b_1^i, \dots, b_{k^i}^i\} \setminus \{a_1^i, \dots, a_{k^i}^i\}$ (ignoring the dummy nodes). The change in welfare is given by $\sum_i v^i(S_i) - v^i(S_i \cup \{b_1^i, \dots, b_{k^i}^i\} \setminus \{a_1^i, \dots, a_{k^i}^i\})$ while the sum of weights of edges in the cycle is given by $\sum_i \sum_{(a,b) \in M^i} v^i(S_i) - v^i(S_i \cup b \setminus a)$. In general, those two quantities are different, so the fact that the cycle has negative weight is not enough to guarantee that performing the exchanges prescribed by it will result in an improvement in welfare.

Murota shows in [22] that if the cycle has minimal cardinality, however, then the total weight of the cycle is equal to the change in welfare by performing the exchanges. Formally:

THEOREM 10.1. *Given gross substitute valuations v^1, \dots, v^m and an allocation S_1, \dots, S_m , if C is a negative weight cycle in the graph defined in Lemma 9.2 and C has the minimum number of edges among all negative cycles, then if $M^i = \{(a, b) \in C; a \in S'_i\}$, $A^i = \{a \in [n]; (a, b) \in M^i\}$, $B^i = \{b \in [n]; (a, b) \in M^i\}$, then the total weight of the cycle corresponds to the change in welfare by performing the exchanges*

prescribed by it:

$$\sum_i v_i(S_i) - v_i(S_i \cup B_i \setminus A_i) = \sum_i \sum_{(a,b) \in M^i} v_i(S_i) - v_i(S \cup b \setminus a)$$

The proof relies of the following Lemma on gross substitute valuations:

LEMMA 10.2. *Given a gross substitute valuation v , a set S , $A = \{a_1, \dots, a_k\} \subseteq S$, $B = \{b_1, \dots, b_k\} \subseteq [n] \setminus S$, 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 $M = \{(a_1, b_1), \dots, (a_k, b_k)\}$ is the unique minimum weight matching in the graph, then:*

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

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 $(a_1, b_1), \dots, (a_{k-1}, b_{k-1})$ 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:

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

In order to $(a_1, b_1), \dots, (a_{k-1}, b_{k-1})$ 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.

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

where the (*) inequality follows from Lemma 4.3.

Now, given a matching \tilde{M} different then $(a_1, b_1), \dots, (a_{k-1}, b_{k-1})$ 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 the 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 than 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 4.3 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 $v(S \cup b_i \setminus a_i) + v(S \cup b_k \setminus a_k) > v(S \cup b_k \setminus a_i) + v(S \cup b_i \setminus a_k)$.

□

Proof. [of Theorem 10.1] From the previous lemma, it is enough to show that if M^i is the unique minimum weighted matching in the bipartite graph with left nodes A^i , right nodes B_i and edge weights $w_{a_i b_j}$. Assume that there is an alternative perfect matching $M' = \{(a_{j_1}^i, b_{j_2}^i), (a_{j_2}^i, b_{j_3}^i), \dots, (a_{j_k}^i, b_{j_1}^i)\}$ with total weight no larger than the original one. Consider now k cycles in graph G where the t -th cycle C_t is formed by edge $(a_{j_t}^i, b_{j_{t+1}}^i)$ and the path from $b_{j_{t+1}}^i$ to $a_{j_t}^i$ in the original cycle C . Each cycle C_i is either C or is a cycle with smaller number of edges. By a simple counting argument, there exists an integer ℓ such that the multiset union of cycles $\{C_i; C_i \neq C\}$ has ℓ copies of each edge in $C \setminus M^i$, $\ell - 1$ copies of each edge in M^i and one copy of each edge in M' . Therefore, the sum of weights of such cycles is at most ℓ times the sum of weights in C and therefore negative. Then there must exist some cycle $C_i \neq C$ of negative weight, contradicting that C has minimal number of edges among all negative cycles. □

The previous discussion suggests an algorithm that strictly improves an allocation, but doesn't guarantee polynomial runtime. A careful choice of cycles following the approach suggested by Zimmermann [28] for the submodular flow problem, is enough to make the algorithm run in strongly polynomial time. We say that a cycle is of *minimum mean weight* if the total weight divided by the number of edges in the cycle is minimized. Such cycle can be found in polynomial time using an algorithm by Megiddo [19].

THEOREM 10.3 (Murota [22]). *The algorithm that finds a minimum weight mean cycle with minimum number of edges among such cycles (Algorithm 4) is a strongly polynomial time algorithm for the gross substitute welfare problem.*

Algorithm 4 Minimum mean weight cycle cancelling

Input: gross substitute valuations $v^1, \dots, v^m : 2^{[n]} \rightarrow \mathbb{R}_+$

Initialize with an arbitrary partition S_1, \dots, S_m of $[n]$

Define G implicitly as the graph in Lemma 9.2

while G has negative weight cycles

find a minimum mean weight cycle C with minimal number of edges

let $A_i = \{a; (a, b) \in C, a \in S_i\}$ and $B_i = \{b; (a, b) \in C, a \in S_i\}$

update $S_i = S_i \cup B_i \setminus A_i$ for all i .

REFERENCES

- [1] L. Ausubel and P. Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1), 2002.
- [2] D. P. Bertsekas. The auction algorithm: a distributed relaxation method for the assignment problem. *Ann. Oper. Res.*, 14(1-4):105–123, June 1988.
- [3] S. Bikhchandani and J. W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74(2):385–413, June 1997.
- [4] M. Bing, D. J. Lehmann, and P. Milgrom. Presentation and structure of substitutes valuations. In *ACM Conference on Electronic Commerce*, pages 238–239, 2004.
- [5] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–72, August 1986.
- [6] A. Dress and W. Terhalle. Rewarding maps: On greedy optimization of set functions. *Advances in Applied Mathematics*, 16(4):464 – 483, 1995.
- [7] A. Dress and W. Terhalle. Well-layered maps a class of greedily optimizable set functions. *Applied Mathematics Letters*, 8(5):77 – 80, 1995.
- [8] A. W. Dress and W. Wenzel. Valuated matroids: a new look at the greedy algorithm. *Applied Mathematics Letters*, 3(2):33 – 35, 1990.
- [9] S. Fujishige. A primal approach to the independent assignment problem. *Journal of the Operations Research Society of Japan*, 20(1):1–15, mar 1977.
- [10] S. Fujishige and Z. Yang. A note on kelso and crawford’s gross substitutes condition. *Math. Oper. Res.*, 28(3):463–469, July 2003.
- [11] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, July 1999.
- [12] F. Gul and E. Stacchetti. The english auction with differentiated commodities. *Journal of Economic Theory*, 92(1):66–95, May 2000.
- [13] 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.
- [14] J. Kelso, Alexander S and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50(6):1483–1504, November 1982.
- [15] B. Korte, L. Lovász, and R. Schrader. *Greedoids*. Algorithms and Combinatorics. Springer-Verlag, 1991.
- [16] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955.
- [17] E. Lawler. *Combinatorial optimization: networks and matroids*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 1976.
- [18] B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.
- [19] N. Megiddo. Combinatorial optimization with rational objective functions. In *STOC*, pages 1–12, 1978.
- [20] K. Murota. Convexity and steinitz’s exchange property. *Advances in Mathematics*, 124(2):272 – 311, 1996.
- [21] K. Murota. Valuated matroid intersection i: Optimality criteria. *SIAM J. Discrete Math.*, 9(4):545–561, 1996.
- [22] K. Murota. Valuated matroid intersection ii: Algorithms. *SIAM J. Discrete Math.*, 9(4):562–576, 1996.
- [23] K. Murota and A. Shioura. M-convex function on generalized polymatroid. *Mathematics of Operations Research*, 24(1):pp. 95–105, 1999.
- [24] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *J. Economic Theory*, 129(1):192–224, 2006.
- [25] J. Oxley. *Matroid theory*. Oxford Graduate Texts in Mathematics Series. Oxford University Press, Incorporated, 1992.
- [26] H. Reijnierse, A. v. Gellekom, and J. A. M. Potters. Verifying gross substitutability. *Economic Theory*, 20(4):pp. 767–776, 2002.
- [27] 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.
- [28] U. Zimmermann. Negative circuits for flows and submodular flows. *Discrete Applied Mathematics*, 36(2):179–189, 1992.