The Invisible Hand of Dynamic Market Pricing

Vincent Cohen-Addad [*]
vcohen@di.ens.fr

Alon Eden [†§]
alonarden@gmail.com

Michal Feldman [‡§]
michal.feldman@cs.tau.ac.il

Amos Fiat [†¶]
fiat@tau.ac.il

November 19, 2015

Abstract

Walrasian prices, if they exist, have the property that one can assign every buyer some bundle in her demand set, such that the resulting assignment will maximize social welfare. Unfortunately, this assumes carefully breaking ties amongst different bundles in the buyer demand set. Presumably, the shopkeeper cleverly convinces the buyer to break ties in a manner consistent with maximizing social welfare. Lacking such a shopkeeper, if buyers arrive sequentially and simply choose some arbitrary bundle in their demand set, the social welfare may be arbitrarily bad. In the context of matching markets, we show how to compute dynamic prices, based upon the current inventory, that guarantee that social welfare is maximized. Such prices are set without knowing the identity of the next buyer to arrive. We also show that this is impossible in general (e.g., for coverage valuations), but consider other scenarios where this can be done.

[*]Ecole normale supérieure, Paris, France
[†]Tel-Aviv University, Israel
[‡]Tel Aviv University and Microsoft Research, Israel
[§]The work of M. Feldman and A. Eden was partially supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement number 337122.
[¶]This work was done in part while the A. Fiat was visiting the Simons Institute for the Theory of Computing.

1 Introduction

A remarkable property of Walrasian pricing is that it is possible to match buyers to bundles, such that every buyer gets a bundle in her demand set (i.e., a set of items S maximizing $v_i(S) - \sum_{j \in S} p_j$), and the resulting allocation maximizes the social welfare, $\sum_i v_i(S_i)$ (S_i being the bundle allocated to buyer i). However, Walrasian prices cannot coordinate the market alone; it is critical that ties be broken appropriately, in a coordinated fashion.

Consider the following scenario: two items, a and b, and two unit demand buyers, Alice and Bob. Alice has value R for item a and value one for item b, Bob has value one for each of the two items a and b. There are many Walrasian pricings in this setting, for example a price of $R-1$ for item a and 0 for item b. Indeed, assigning item a to Alice and item b to Bob under these prices maximizes simultaneously the individual utility of each buyer and the social welfare.

However, in real markets, buyers often arrive sequentially, in some unknown order, and get no guidance as to how to break ties. For these prices, ($p(a) = R-1$ and $p(b) = 0$), if Bob arrives first then he will indeed choose item b, leaving item a for Alice to purchase, resulting in a social welfare maximizing allocation. If, however, Alice arrives first, she has equal utility ($= 1$) for both a and b and may select item b, so Bob will walk away without purchasing any item, which results in social welfare 1, compared with the optimal social welfare of $R+1$. We furthermore remark that setting prices of $p(a) = R$ and $p(b) = 1$, which are also Walrasian prices, could result in both Alice and Bob walking away, and resulting in zero social welfare.

One may suspect that we choose the wrong Walrasian pricing. It is known that in matching markets the minimal Walrasian prices coincide with VCG payments [Leo83]. In this example the minimal Walrasian prices are to charge zero for both item a and item b. Indeed, if Alice arrives first, she will choose item a, and when Bob arrives he will choose item b, and this is the social welfare maximizing allocation. However, if Bob arrives first, he will be indifferent between the two items and may choose item a — again — this achieves a social welfare of 2 compared with the optimal social welfare of $R+1$. Moreover, there exist markets that admit unique Walrasian prices, yet may achieve zero welfare. For example, consider a single item valued at 1 by both Alice and Bob. The unique Walrasian price is 1, which may result in both buyers walking away without purchasing the item.

In fact, we can show that no static prices (and thus no Walrasian prices) can give more than 2/3 of the social welfare for buyers that arrive sequentially. Consider unit demand buyers Alice, Bob, and Carl, and items a, b, and c. Alice values a and b at one, and has zero value for c, symmetrically, Bob values b and c at one and a at zero, and Carl values c and a at one, and b at zero. A two line proof shows that no static pricing scheme, $p(a)$, $p(b)$, and $p(c)$ can achieve more than 2/3 of the optimal social welfare. Assume all prices are strictly less than one, and assume, without loss of generality, that $p(a) \geq p(b) \geq p(c)$. Now, Alice arrives and chooses item b, Carl arrives and chooses item c, and finally Bob arrives — but there are no items left for which Bob has a non zero valuation. Note that if $p(a) \geq 1$ then item a will not be sold as whomever is to buy it may decide simply to walk away, the same holds for items b and c so assuming that all prices are strictly less than one holds without loss of generality, given that one assumes that the prices achieve $\geq 2/3$ of the optimal social welfare.

However, consider the following twist, which changes the prices after the first buyer arrives. In the scenario above, when Alice arrives first and chooses (without loss of generality) item a, change the prices so that Bob will choose b and Carl will choose c. This is easily done by setting new prices

$p'(b) < p'(c)$. Irrespective of whomever arrives after Alice, the prices will ensure that all items get sold and social welfare be maximized.

Obtaining optimal social welfare is trivial via dynamic pricing if the pricing mechanism knew which buyer was to arrive next. The dynamic pricing mechanism could make use of infinite prices to reduce the choices available to incoming buyer so that only a bundle consistent with optimal social welfare can be selected. The key difficulty arises because the prices need be set *before* the preferences of the next buyer to arrive are known.

Thus, this paper studies the issues of static and dynamic pricing for sequentially arriving buyers. Our main result is the following:

Main Theorem: For any matching market (*i.e.*, unit demand valuations), we give a poly-time dynamic pricing scheme that achieves the optimal social welfare, for any arrival order and irrespective of any tie breaking chosen by the buyers.

We show that the existence of Walrasian prices does not, by itself, imply that there exist dynamic pricing schemes that optimize social welfare. In particular, we give an example (Section 4) of a market with coverage valuations (a strict subclass of submodular valuations), which has a unique optimal solution, and where Walrasian prices do exist, and yet no dynamic pricing scheme (static or dynamic) can get the optimal social welfare.

We offer some remedies for this impossibility result.

- We show that a market with gross substitutes valuations that has a unique optimal allocation always admits a *static item pricing* scheme that achieves the optimal welfare (Section 6)[1]
- Moreover, while full efficiency is in general impossible, we argue that for *any* profile of valuations, there exists a static pricing scheme that achieves at least a half of the optimal social welfare. This result can be viewed as a generalization of the Combinatorial Walrasian Equilibrium of [FGL13]. In fact we adapt the static bundle prices computed in [FGL15] for Bayesian agents to achieve the one half guarantee of the optimal social welfare, for any class of valuations.
- We identify additional classes of valuations that admit dynamic pricing schemes that obtain the optimal social welfare: (1) where buyer i seeks up to k_i items, and valuations depend on the item, and (2) for superadditive valuations.

The following remark is in order. Gross substitutes valuations are known to be the frontier for the guaranteed existence of a Walrasian equilibrium [GS99]. They are also the frontier with respect to computational tractability [NS06]: one can compute the allocation that maximizes social welfare in polynomial time. *Are gross substitutes valuations also the frontier for achieving optimal welfare via a dynamic pricing scheme?* More formally:

Main Open Problem: Does any market with gross substitutes valuations have a dynamic pricing scheme that achieves optimal social welfare?

[1]In a talk by Aaron Roth at the Simons Institute on October 16, 2015, attended by some of the authors, he mentioned that unique optimal allocations for gross substitute allocations allow "no conflict pricing", this immediately implies that such prices will give optimal social welfare for sequentially arriving buyers. This was obtained independently by us.

1.1 Related Work

This paper combines issues of online computation and markets.

Walrasian equilibrium, where prices are such that optimal social welfare is achieved, and the market clears, given appropriate tie-breaking of preferences in the demand set dates back to 1874 [Wal74]. The existence of Walrasian prices for matching markets and more generally for gross substitutes valuations appears in [KJC82, GS99]. We give a definition of these valuations in Section 6. Competitive analysis of online matchings were first studied in [KVV90] where a randomized $1 - 1/e$ approximation to the size of the maximal matching was given.

The use of bundle pricing for Combinatorial Walrasian Equilibria (and no envy amongst buyers), while achieving one half of the social welfare, was given in [FGL13]. The use of static item prices for buyers arriving via a Bayesian process, with XOS valuations, which also achieves 1/2 of the optimal social welfare, was given in [FGL15].

The performance of posted price mechanisms was also studied under the objective of maximizing revenue in Bayesian settings, where it was shown to extract a constant fraction of the optimal revenue for single item settings [BH08] as well as for matching markets [CHK07, CHMS10, CMS10].

More generally, some motivation for this paper is to find more applications of the framework of pricing dynamic decisions [CEFJ15], a general approach to setting dynamic prices on future selfish decisions so as to achieve some predefined goal. In particular, this has been done in the context of minimizing the costs of selfish metrical matchings, selfish metrical task systems, and the selfish k-server problem.

1.2 The Structure of this Paper

In section 2 we describe several types of pricing schemes for sequential buyers, static and dynamic, item prices and bundle prices.

In section 3 we give a dynamic pricing scheme that achieves optimal social welfare, irrespective of how agents break ties, and for any order of arrival. We include a running example to help in clarifying the concepts and algorithms involved.

In section 4 we show that dynamic pricing schemes cannot achieve optimal social welfare even if all of the following hold simultaneously: (1) Walrasian prices exist, (2) The socially optimal allocation is unique, and (3) The valuation is a coverage valuation.

In section 5 we argue that the ideas in [FGL13, FGL15] allow us to compute static prices that achieve 1/2 of the optimal social welfare, for any order of arrival, and any valuation.

In section 6 we show how to compute static item prices that achieve optimal social welfare for sequentially arriving buyers if the valuation class is gross substitutes and the optimal allocation is unique.

In section 7 we show how to compute static bundle prices that achieve optimal social welfare for sequentially arriving buyers if the valuation class is super additive.

In section A we show how to compute dynamic bundle prices that achieve optimal social welfare for sequentially arriving buyers if the valuation class is such that bidder i seeks up to k_i items, and the item values depend only on the item.

2 Model and Preliminaries

Our setting consists of a set I of m indivisible items and a set of n buyers that arrive sequentially in some arbitrary order.

Each buyer has a valuation function $v_i : 2^I \to \Re_{\geq 0}$ that indicates his value for every set of objects, and a buyer valuation profile is denoted by $\mathbf{v} = (v_1, \ldots, v_n)$. We assume valuations are monotone non-decreasing and normalized (i.e., $v_i(\emptyset) = 0$). We use $v_i(A|B) = v_i(A \cup B) - v_i(B)$ to denote the marginal value of bundle A *given* bundle B. An *allocation* is a vector of disjoint sets $\mathbf{x} = (x_1, \ldots, x_n)$, where x_i denotes the bundle associated with buyer $i \in [n]$ (note that it is not required that all items are allocated). The *social welfare* (SW) of an allocation $\mathbf{x}$ is $\mathsf{SW}(\mathbf{x}, \mathbf{v}) = \sum_{i=1}^{n} v_i(x_i)$, and the optimal welfare is denoted by $\mathrm{OPT}(\mathbf{v})$. When clear from context we omit $\mathbf{v}$ and write SW and OPT for the social welfare and optimal welfare, respectively.

An *item pricing* is a function $\mathbf{p} : I \to \Re^{\geq 0}$ that assigns a price to every item. The price of item j is denoted by $p(j)$. Given an item pricing, the utility that buyer i derives from a set of items S is $u_i(S, \mathbf{p}) = v_i(S) - \sum_{j \in S} p(j)$. The *demand correspondence* $D_i(I, \mathbf{p})$ of buyer i contains the sets of objects that maximize buyer i's utility; i.e., $D_i(I, \mathbf{p}) = \mathrm{argmax}_{S \subseteq I} u_i(S, \mathbf{p})$.

A *bundle pricing* is a tuple $(\mathcal{B}, \mathbf{p})$, where $\mathcal{B} = \{B_1, \ldots, B_k\}$ is a partition of the items into bundles (where $\bigcup_i B_i = I$ and for every $i \neq j$, $B_i \cap B_j = \emptyset$), and $p : \mathcal{B} \to \Re^{\geq 0}$ is a function that assigns a price to every bundle in $\mathcal{B}$. The price of bundle B_j is denoted $p(B_j)$. Given a bundle pricing $(\mathcal{B}, \mathbf{p})$, the utility that buyer i derives from a set of bundles S is $u_i(S, \mathbf{p}) = v_i(S) - \sum_{B_j \in S} p(B_j)$. The *demand correspondence* $D_i(I, \mathbf{p})$ of buyer i contains the sets of bundles that maximize buyer i's utility; i.e., $D_i(I, \mathbf{p}) = \mathrm{argmax}_{S \subseteq I} u_i(S, \mathbf{p})$.

We consider several types of pricing schemes: *static item pricing*, *dynamic item pricing*, *static bundle pricing*, and *dynamic bundle pricing*.

In static pricing schemes, prices are assigned (to items or bundles) initially, and never change then. In contrast, in dynamic pricing schemes, new (item or bundle) pricing may be set before the next buyer arrives. Item pricing schemes assign prices to items, whereas bundle pricing schemes can partition the items to bundles and assign prices to bundles that are elements of the partition. Thus, the four types of pricing schemes are described as follows.

Static Item Pricing Scheme:

1. Item prices, $\mathbf{p}$, are determined once and for all.
2. Buyers arrive at some arbitrary order, the next buyer to arrive chooses a bundle in her demand set from among the items not already allocated (and pays the sum of the corresponding prices).

Static Bundling Pricing Scheme:

1. Bundles, and their prices, $(\mathcal{B}, \mathbf{p})$, are determined once and for all.
2. Buyers arrive at some arbitrary order, the next buyer to arrive chooses a set of bundles in her demand set from amongst the bundles not already allocated (and pays the sum of the corresponding prices).

Dynamic Item Pricing Scheme:

- Before buyer $t = 1, \ldots, n$ arrives (and after buyer $t - 1$ departs, for $t > 1$):
  1. Item prices, $\mathbf{p}_t$, are set (or reset) before buyer t arrives, prices are set for those items that have not been purchased yet.

2. When buyer t arrives she purchases a set of items S in her demand from among the items not already allocated (and pays the sum of the corresponding prices according to $\mathbf{p}_t$).

Dynamic Bundle Pricing Scheme:

- Before buyer $t = 1, \ldots, n$ arrives (and after buyer $t-1$ departs, for $t > 1$):
  1. A partition into bundles and bundle prices, $(\mathcal{B}, \mathbf{p}_t)$, is determined for the items that have not been purchased yet.
  2. When buyer t arrives she purchases a set of bundles S in her demand set from among the bundles on sale (and pays the sum of the corresponding prices according to $\mathbf{p}_t$).

We say that a pricing scheme achieves optimal (respectively, α-approximate) social welfare if for any arrival order and any manner in which agents may break ties, the obtained social welfare is optimal (resp., at least α fraction of the optimal welfare).

3 Optimal Dynamic Pricing Scheme for Matching Markets

In this section we consider matching markets. Every agent seeks one item, and may have different valuations for the different items. Whereas this setting admits Walrasian prices, such prices are not applicable to the setting where agents arrive sequentially, in an unknown order, and choose an arbitrary item in their demand set.

We now describe a dynamic item pricing scheme for matching markets that maximizes social welfare — the sum of buyer valuations for their allocated items is maximized. The process we consider is as follows:

- The valuations of the buyers are known.
- The buyers arrive in some arbitrary order unknown to the pricing scheme.
- Prices are posted, they may change after a buyer departs but cannot depend upon the next buyer.

[**Running example**] To illustrate the process, we consider a running example of a matching market, buyers Alice, Bob, Carl, and Dorothy, items a, b, c and d. The valuations are given in Figure 4(a), where squares represent buyers, circles represent items, and A, B, C, D stand for Alice, Bob, Carol and Dorothy. The minimal Walrasian pricing is $p(a) = 1, p(b) = 7, p(c) = 7, p(d) = 0$. Under the minimal Walrasian pricing, or any static pricing, unless ties are broken in a particular way, sequential arrival of buyers will not produce optimal social welfare (see Lemma B.1).
An example of the use of dynamic pricing that follows from our dynamic pricing scheme is given in Figure 4. Every row represent a phase in the process, where a single buyer arrives. The LHS graph in every row represents the valuations of the remaining buyers and items, thinks edges represent a maximal matching. The RHS graph represents the graph of edges, upon which prices are calculated by Algorithm Price-Items. Directed cycles of length 0 (if any) are represented by think edges. The arriving buyer along with the items they pick are specified in the right column.

The input consists of the graph $G = (N, I, \mathbf{v})$. G is a complete bipartite weighted graph, where N is the set of agents, I is the set of items, and for every agent $a \in N$ and item $b \in I$, the weight of an edge $\langle a, b\rangle$ is the value that agent a gives item b, $v_a(b)$ ($v_a : I \to \Re_{\geq 0}$ is the valuation function for agent a).

Without loss of generality, one may assume that in G we have that $|I| \geq |N|$, otherwise, we add dummy vertices to the I with zero weight edges to the vertices of the N side until $|I| = |N|$. OPT is the weight of the maximum weighted matching in G (alternatively, the optimal social welfare). Let $M \subseteq N \times I$ be some matching in G, we define $\mathsf{SW}(M) = \sum_{(a,b)\in M} v_a(b)$ to be a function that takes a matching and returns the social welfare (value) of the matching.

We now continue to describe the dynamic pricing scheme. At time $t \in 0, \ldots, |N|$ (after the t-th agent departs), we define the following:

- $M_t \subseteq N \times I$ is the partial matching consisting of [a subset] of the first t agents to arrive, and the item of their choice, amongst the items available for sale upon arrival. The size of M_t may be less than t as not all buyers may be matched as their demand set may be empty when they arrive.
- $N_t \subseteq N$ and $I_t \subseteq I$ are the first t agents to arrive and the items matched to them in the matching M_t.
- $N_{>t} = N \setminus N_t$ and $I_{>t} = I \setminus I_t$ are the remaining agents (to arrive at some time $> t$) and the items remaining after the departure of the t-th agent. Define $G_{>t}$ to be the graph G where agents N_t and items I_t have been discarded. *I.e.*, $G_{>t} = (I_{>t}, N_{>t}, \mathbf{v})$.
- We define $p_{t+1} : I_{>t} \to \Re_{\geq 0}$ to be the prices set by the dynamic pricing scheme after the departure of agent t (but before the arrival of agent $t + 1$).

To compute the function p_{t+1} we first construct a so-called "relation graph", and then perform various computations upon it. The vertices of the relation graph are all goods yet unsold, $I_{>t}$, the edges and their weights are as follows:

1. Compute $M_{>t} \subseteq I_{>t} \times N_{>t}$, a maximum weight matching of the graph $G_{>t}$ which matches **all** vertices of $I_{>t}$.[2] For every item $b \in I_{>t}$, let $v_{>t}(b)$ denote the value of item b to the agent matched to item b in the matching $M_{>t}$.

2. The edges of $R_{>t}$, denoted by $E_{>t}$, are a clique on the vertices $I_{>t}$, and their weights $W_{>t} : E_{>t} \to \Re$ are computed as follows: Let $M_{>t}$ be a maximum weight matching of remaining goods and agents as defined above. For every pair $(a, b) \in M_{>t}$, and for every $b' \in I_{>t} \setminus \{b\}$ create an edge $\langle b, b'\rangle$. The weight of the edge $\langle b, b'\rangle$,

$$W_{>t}(\langle b, b'\rangle) = v_a(b) - v_a(b').$$

[Running example] The initial graph $G_{>0}$ of our running example is given in Figure 4(a), where a maximal matching $M_{>0}$ is indicated by the thick edges. The graph $R_{>0}$ is given in Figure 4(b). For example, the weight of the edge $\langle a, b\rangle$ is $v_{Alice}(a) - v_{Alice}(b) = -6$.

We give the following structural property of $R_{>t}$:

[2] Note that such a maximum weight matching exists because initially $|N| \leq |I|$, and since every agent takes at most one item, $|N_{>t}| \leq |I_{>t}|$ continues to hold. Since all edge weights are non-negative, and $G_{>t}$ is a complete bipartite graph, every maximum weight matching can be extended to produce a matching with the same weight which matches all of the vertices in $I_{>t}$.

Lemma 1 *There are no directed cycles of negative weight in $R_{>t}$.*

Proof: Assume there exists a negative cycle of length ℓ. Assume the cycle is comprised of $\langle b_1, b_2\rangle, \langle b_2, b_3\rangle, \ldots, \langle b_{\ell-1}, b_\ell\rangle, \langle b_\ell, b_1\rangle$. This cycle corresponds to a cycle of alternating edges in $G_{>t}$ $(b_1, a_1)\,(a_1, b_2)\,,(b_2, a_2)\ldots(a_{\ell-1}, b_\ell)\,,(b_\ell, a_\ell)\,,(a_\ell, b_1)$, where for every $j \in \{1, \ldots, \ell\}$, $(b_j, a_j) \in M_t$ and $(a_j, b_{j+1}) \notin M_t$.

For ease of notation, we define $\ell + 1 = 1$. According to the definition of weights in $R_{>t}$, we know that

$$\sum_{j=1}^{\ell} W_{>t}(\langle b_j, b_{j+1}\rangle) = \sum_{j=1}^{\ell} \left(v_{a_j}(b_j) - v_{a_j}(b_{j+1})\right) < 0,$$

and therefore, $\sum_{j=1}^{\ell} v_{a_j}(b_{j+1}) > \sum_{j=1}^{\ell} v_{a_j}(b_j)$. We get that the matching M', which is constructed by removing the set $\{(b_j, a_j)\}_{j\in 1,\ldots,\ell}$ from $M_{>t}$ and adding the set $\{(b_{j+1}, a_j)\}_{j\in 1,\ldots,\ell}$, is of larger weight, in contradiction to $M_{>t}$ being a maximum weight matching.

■

We now process the relation graph $R_{>t}$:

1. Let Δ be the smallest total weight of a cycle with strictly positive total weight in $R_{>t}$, and let $\epsilon = \frac{\Delta}{|I_{>t}|+1}$. Mark all edges in $E_{>t}$ that take part in **some** directed cycle of weight 0 in $R_{>t}$. Delete all marked edges. For every remaining edge e, set $W'_{>t}(e) = W_{>t}(e) - \epsilon$. Let $R'_{>t} = (I_{>t}, E'_{>t}, W'_{>t})$ be the resulting graph.

2. Find a solution to the set of equations in Figure 1 by running algorithm Price-Items (see Figure 2) with $R'_{>t}$ as the input graph. Set $p_{t+1} = p$ where p is the output of Price-Items.

To show that indeed, $R'_{>t}$ can be used as an input for Price-Items, we show the following:

Lemma 2 *All the directed cycles in $R'{}_{>t}$ are strictly positive.*

Proof: Let $\widetilde{R}$ be the graph which is obtained from $R_{>t}$ by removing all the edges that take part in a directed cycle of weight 0. Since according to Lemma 1, $R_{>t}$ has no negative weight cycles, all the cycles in $\widetilde{R}$ are of strictly positive weight. By the definition of Δ, every simple cycle has a weight of at least Δ. $R'{}_{>t}$ is constructed by taking $\widetilde{R}$ and decreasing all the edge weights by $\epsilon = \frac{\Delta}{|I_{>t}|+1}$. Therefore, the weight of every simple cycle in $\widetilde{R}$ could have decreased by no more than $|I_{>t}|\,\epsilon < \Delta$, which means that all the cycles in $R'{}_{>t}$ are of strictly positive weight. ■

[**Running example**] In Figure 4(b), the thick edges form a directed cycle of weight 0. We remove these edges and subtract ϵ from every remaining edge. We then run Algorithm Price-Items on the obtained graph, which gives the prices presented in red next to each item in Figure 4(b). In this case, the only negative edge (after removing the cycle of length 0) is the edge $\langle d, a\rangle$, whose price is set to $-W'(\langle d, a\rangle) = -(-1-\epsilon) = 1+\epsilon$. Since all other shortest paths are positive, prices of other items do not change (recall the new price is the maximum between the old price and the negation of the shortest path). When Alice arrives, she picks the unique item in her demand set — item a. Similarly, graphs $G_{>t}, R_{>t}$ of all iterations $t = 0, 1, 2, 3$ are demonstrated in Figure 4(c)-(h).

$$\forall b \in I_{>t} \qquad p(b) \geq 0 \tag{1}$$
$$\forall \langle b_1, b_2 \rangle \in E'_{>t} \qquad p(b_1) - p(b_2) < W_{>t}(\langle b_1, b_2 \rangle) \tag{2}$$
$$\forall b \in I_{>t} : v_{>t}(b) > 0 \qquad p(b) < v_{>t}(b) \tag{3}$$

Figure 1: The set of equations that ensures every greedy agent would choose an edge of some maximum weight matching.

Price-Items
Input: A directed graph $G = (I, E, W)$ where all cycles are strictly positive.
Output: a pricing function $p : I \rightarrow \Re_{\geq 0}$ such that $p(b') - p(b) \geq -W(\langle b, b' \rangle)$ for every $\langle b, b' \rangle \in E$.

1. Set $p(b) \leftarrow 0$ for every $b \in I$.
2. Run *all-pairs-shortest-paths* on G (there are no negative cycles in G). For every $b, b' \in I$, let $d(b, b')$ denote the length of the shortest path from b to b'.
3. For every $b \in I$:
   (a) For every $b' \in I$, set $p(b') \leftarrow \max\{p(b'), -d(b, b')\}$.

Figure 2: Pricing algorithm.

Consider a directed edge $\langle b_1, b_2 \rangle$ and some cycle it belongs to. The edge $\langle b_1, b_2 \rangle$ came about because we choose a maximal matching where item b_1 was assigned to some buyer a, whereas b_2 was not. If all such cycles have strictly positive total weight, then the edge weights, and the associated prices computed via Price-Items, ensure that agent a prefers b_1 to b_2, effectively removing choices for "wrong" tie breaking. Contrawise, if the edge $\langle b_1, b_2 \rangle$ does belongs to some cycle of total weight zero, this implies that the maximum matching is not unique. Ergo, whenever some item along this cycle is first chosen, it is still possible to extend the matching to a maximal weight matching. This is exactly where the dynamic pricing creeps in, subsequent to this symmetry breaking, new prices have to be computed to avoid wrong tie breaking decisions.

We now show that setting prices that satisfy the constraints in Figure 1 ensures that after all agents arrive, the social welfare achieved is maximized.

Theorem 3.1 *A dynamic pricing scheme which calculates prices satisfying the constraints presented in Figure 1 achieves optimal social welfare (a maximum weight matching of G).*

Proof: Recall that M_t is the matching which results from the first $t \in \{0, 1, \ldots, |N|\}$ agents taking an item which maximizes their utility and that $G_{>t}$ is the graph of the remaining agents and items after the first t agents arrived and purchased some items. Let $M_{>t}$ be a maximum weight matching of $G_{>t}$, where $M_{>0}$ is a matching that maximizes the social welfare of all the agents, and $M_{>|N|} = \emptyset$. We prove by induction that for every $i \in \{0, 1, \ldots, |N|\}$, $\mathsf{SW}(M_t) + \mathsf{SW}(M_{>t}) = OPT$. It follows that the matching $M_{|N|}$ yields optimal social welfare.

For $t = 0$, this claim trivially holds since $\mathsf{SW}(M_{>0}) = OPT$. Assume that for some $t - 1$, $\mathsf{SW}(M_{t-1}) + \mathsf{SW}(M_{>t-1}) = OPT$. Let $M_{>t}$ be the maximum weight matching we compute at step 1. of the pricing scheme. When agent t arrives, consider the following cases:

- Agent t does not take any item. From the constraints of type (3), the only case where an has no positive utility from any item is if she is matched to an item in M_{t-1} with an edge of weight 0. In this case, $\mathsf{SW}(M_t) = \mathsf{SW}(M_{t-1})$, and by taking $M_{>t}$ to be the same matching as $M_{>t-1}$ without the edge the t-th agent is matched to, $\mathsf{SW}(M_{>t}) = \mathsf{SW}(M_{>t-1})$. We get that $\mathsf{SW}(M_t) + \mathsf{SW}(M_{>t}) = \mathsf{SW}(M_{t-1}) + \mathsf{SW}(M_{>t-1}) = OPT$.

- Agent t takes the item which she is matched to in $M_{>t-1}$. Let v be the value of the t-th agent for the item. Clearly, $\mathsf{SW}(M_t) = \mathsf{SW}(M_{t-1}) + v$. By taking $M_{>t}$ to be the same matching as $M_{>t-1}$ without the edge the t-th agent is matched to, we get $\mathsf{SW}(M_{>t}) = \mathsf{SW}(M_{>t-1}) - v$. We get that $\mathsf{SW}(M_t) + \mathsf{SW}(M_{>t}) = \mathsf{SW}(M_{t-1}) + v + \mathsf{SW}(M_{>t-1}) - v = OPT$.

- Agent t $a \in N_{>t-1}$ takes an item $b' \in I_{>t-1}$ which is different than $b \in I_{>t-1}$, the item which she is matched to in $M_{>t-1}$. Therefore,

$$v_a(b') - p_{t-1}(b') \geq v_a(b) - p_{t-1}(b). \tag{4}$$

  Let $\langle b, b' \rangle \in E_{>t-1}$ be the directed edge from b to b' in $R_{>t-1}$. Its weight $W_{>t}(\langle b, b' \rangle) = v_a(b) - v_a(b')$. If $\langle b, b' \rangle$ would have been in $R'_{>t-1}$, then according to constraint (2), we would have had that $p_{t-1}(b) - p_{t-1}(b') < W_{>t}(\langle b, b' \rangle) = v_a(b) - v_a(b')$. Rearranging gives us $v_a(b') - p_{t-1}(b') < v_a(b) - p_{t-1}(b)$, which contradicts (4). Therefore, $\langle b, b' \rangle$ was removed from $R'_{>t-1}$, which can only happen if the edge is part of a directed cycle of weight 0 in $R_{>t-1}$.

  Let $b_0 = b$, $b_1 = b'$ and let $\langle b_0, b_1 \rangle, \langle b_1, b_2 \rangle, \ldots, \langle b_{\ell-1}, b_\ell \rangle, \langle b_\ell, b_0 \rangle$ be a simple directed cycle of length $\ell + 1$ and weight 0 in $R_{>t-1}$ in which $\langle b, b' \rangle$ takes part. This cycle corresponds to a cycle of alternating edges in $G_{>t-1}$,

$$(b_0 = b, a_0 = a) \left(a_0, b_1 = b'\right), (b_1, a_1) \ldots (a_{\ell-1}, b_\ell), (b_\ell, a_\ell), (a_\ell, b_0),$$

  where

$$(b_j, a_j) \in M_{>t-1} \text{ and } (a_j, b_{j+1 \mod \ell}) \notin M_{>t-1} \quad \text{for every } j \in \{0, \ldots, \ell\}.$$

  Since the directed cycle is of weight 0, we get that

$$\sum_{j=0}^{\ell} W_{>t}(\langle b_j, b_{j+1 \mod \ell} \rangle) = \sum_{j=0}^{\ell} \left(v_{a_j}(b_j) - v_{a_j}(b_{j+1 \mod \ell})\right) = 0,$$

  which means that the value of the unmatched edges in the directed cycle, $\sum_{j=0}^{\ell} v_{a_j}(b_{j+1 \mod \ell})$, is equal to the value of the matched edges, $\sum_{j=0}^{\ell} v_{a_j}(b_j)$.

  Let $\widetilde{M}_{>t-1}$ be the matching which is a result of taking $M_{>t-1}$, removing the edges in the set $\{(a_j, b_j)\}_{j \in \{0,1,\ldots,\ell\}}$, and adding the edges of $\{(b_{j+1 \mod \ell}, a_j)\}_{j \in \{0,1,\ldots,\ell\}}$; Note that $(a, b') = (a_0, b_1) \in \widetilde{M}_{>t-1}$. Since the edges we added to $\widetilde{M}_{>t-1}$ are of the same value as the edges we removed, $\mathsf{SW}(\widetilde{M}_{>t-1}) + \mathsf{SW}(M_{t-1}) = \mathsf{SW}(M_{>t-1}) + \mathsf{SW}(M_{t-1}) = OPT$. We define $M_{>t}$ to be a matching comprised of the same edges as $\widetilde{M}_{>t-1}$ except (a, b'). Therefore, $\mathsf{SW}(M_{>t}) = \mathsf{SW}(\widetilde{M}_{>t-1}) - v_a(b')$. Clearly, we have that $\mathsf{SW}(M_t) = \mathsf{SW}(M_{t-1}) + v_a(b')$. We get that $\mathsf{SW}(M_{>t}) + \mathsf{SW}(M_t) = \mathsf{SW}(\widetilde{M}_{>t-1}) - v_a(b') + \mathsf{SW}(M_{t-1}) + v_a(b') = OPT$. This completes the proof of the induction and the theorem.

■

It remains to show that Price-Items satisfies all the constraints in Figure 1. First, we observe that constraints of type (1) are trivially satisfied since all prices are initially set to 0 by Price-Items and prices can only increase.

Observation 3.2 *Price-Items returns an assignment which satisfies constraints of type* (1)*.*

The following property is helpful in proving that constraints of type (2):

Lemma 3 *Let $G = (I, E, W)$ be the input graph of Price-Items and let $p : I \to \Re_{\geq 0}$ be its output. For every $\langle b_1, b_2 \rangle \in E$ we have that $p(b_2) - p(b_1) \geq -W(\langle b_1, b_2 \rangle)$.*

Proof: We first show that after all the iterations of step 3 of Price-Items, for any two vertices $b_1, b_2 \in I$, $p(b_2) - p(b_1) \geq -d(b_1, b_2)$. b_1 was chosen as b at step 3 of a some iteration of the loop. Let b_2 be some vertex reachable from b_1 (otherwise, $d(b_1, b_2) = \infty$ and the claim trivially holds). If $p(b_1) = 0$ after all iterations, then when b_1 was chosen as b in step 3, $p(b_2) \geq -d(b_1, b_2) = p(b_1) - d(b_1, b_2)$, implying that $p(b_2) - p(b_1) \geq -d(b_1, b_2)$. Since $p(b_1)$ stayed the same and $p(b_2)$ did not decrease, the inequality still holds.

If $p(b_1) > 0$, let $\widetilde{b}$ be the vertex which was chosen in step 3 in the iteration where the current $p(b_1)$ was set. At the iteration where the current $p(b_1)$ was set, we have that $p(b_1) = -d(\widetilde{b}, b_1)$ and

$$\begin{aligned} p(b_2) &= -d(\widetilde{b}, b_2) \\ &\geq -(d(\widetilde{b}, b_1) + d(b_1, b_2)) \\ &= p(b_1) - d(b_1, b_2), \end{aligned}$$

where the inequality follows since the shortest path satisfies the triangle inequality. We get that at the iteration where the current $p(b_1)$ was set, $p(b_2) - p(b_1) \geq -d(b_1, b_2)$. Since $p(b_1)$ stayed the same until the current iteration, and $p(b_2)$ did not decrease, the inequality still holds.

Since $d(b_1, b_2) \leq W(\langle b_1, b_2 \rangle)$, we get the desired result. ■

We can now establish that constraints of type (2) hold.

Lemma 4 *Price-Items returns an assignment which satisfies constraints of type* (2)*.*

Proof: By Lemma 3, we get that for a given $\langle b_1, b_2 \rangle \in E'_{>t}$, $p(b_2) - p(b_1) \geq -W'_{>t}(\langle b_1, b_2 \rangle) = -(W_{>t}(\langle b_1, b_2 \rangle) - \epsilon)$. Therefore, $p(b_1) - p(b_2) \leq W_{>t}(\langle b_1, b_2 \rangle) - \epsilon < W_{>t}(\langle b_1, b_2 \rangle)$, as desired. ■

For establishing that constraints of type (3) are met by the prices $p(b)$'s computed by Price-Items, we need the following Lemma.

Lemma 5 *Let b_ℓ be some vertex with $p(b_\ell) > 0$. Let b_0 be the vertex chosen at the iteration of the loop in Price-Items where $p(b_\ell)$ was set, and let $\langle b_0, b_1 \rangle, \langle b_1, b_2 \rangle, \ldots, \langle b_{\ell-1}, b_\ell \rangle$ be a shortest path from b_0 to b_ℓ. For every $i \in \{0, 1, \ldots, \ell\}$, $p(b_i) = -d(b_0, b_i)$.*

Proof: Let b_i a vertex on the shortest path from b_0 to b_ℓ such that $p(b_i) > -d(b_0, b_i)$ (notice that step $3a$ of Price-Items ensures that $p(b_i) \geq -d(b_0, b_i)$). This can only happen if there exists some $\widetilde{b}$ such that $d(\widetilde{b}, b_i) < d(b_0, b_i)$. Since b_i is on the shortest path from b_0 to b_ℓ, we know that $d(b_0, b_\ell) = d(b_0, b_i) + d(b_i, b_\ell)$. We get that

$$\begin{aligned} d(\widetilde{b}, b_\ell) &\leq d(\widetilde{b}, b_i) + d(b_i, b_\ell) \\ &< d(b_0, b_i) + d(b_i, b_\ell) \\ &= d(b_0, b_\ell), \end{aligned}$$

where the first inequality is due to the triangle inequality. Therefore, when $\widetilde{b}$ is chosen in step 3 of Price-Items, step 3a ensures that $p(b_\ell) \geq -d(\widetilde{b}, b_\ell) > -d(b_0, b_\ell)$. Since b_ℓ was set in the iteration where b_0 was chosen at step 3, we also get that $p(b_\ell) = -d(b_0, b_\ell)$, a contradiction. ■

We get the the following two corollaries.

Corollary 3.3 $p(b_0) = 0$.

Corollary 3.4 *For every* $i \in \{0, 1, \ldots, \ell - 1\}$, $p(b_i) - p(b_{i+1}) = W_{>t}(\langle b_i, b_{i+1} \rangle) - \epsilon$.

Proof: Since every subset of a shortest path is also a shortest path, we get that $d(b_0, b_{i+1}) = d(b_0, b_i) + W'_{>t}(\langle b_i, b_{i+1} \rangle)$. From Lemma 5, we get that $p(b_i) = -d(b_0, b_i)$ and

$$\begin{aligned} p(b_{i+1}) &= -d(b_0, b_{i+1}) \\ &= -d(b_0, b_i) - W'_{>t}(\langle b_i, b_{i+1} \rangle) \\ &= p(b_i) - (W_{>t}(\langle b_i, b_{i+1} \rangle) - \epsilon), \end{aligned}$$

where the last equality follows by the definition of $W'_{>t}$. ■

We now prove that all the constraints of type (3) are met.

Lemma 6 *For every* $b \in I_{>t}$ *which is matched in* $M_{>t}$ *by a non-zero weight edge,* $p(b) < v_{>t}(b)$.

Proof: Assume for the purpose of reaching a contradiction that there exists some $b = b_\ell$ which is matched in M_t via an edge of strictly positive weight for which where $p(b) \geq v_{>t}(b)$. Let b_0 be the vertex that was selected at the iteration of the loop in Price-Items where $p(b)$ was set, and let $\langle b_0, b_1 \rangle, \langle b_1, b_2 \rangle, \ldots, \langle b_{\ell-1}, b_\ell \rangle$ be a shortest path from b_0 to b_ℓ in $R'_{>t}$. According to Corollary 3.4, for every $i \in \{0, 1, \ldots, \ell - 1\}$, $p(b_i) - p(b_i) = W_{>t}(\langle b_i, b_{i+1} \rangle) - \epsilon$. Summing over all i's gives us

$$\sum_{i=0}^{\ell-1} W_{>t}(\langle b_i, b_{i+1} \rangle) = p(b_0) - p(b_\ell) + \ell\epsilon < -p(b) + \Delta, \tag{5}$$

where the inequality stems from the fact that $p(b_0) = 0$ (Corollary 3.3), $b_\ell = b$, $\ell < |I_{>t}|$ and $\epsilon = \frac{\Delta}{|I_{>t}|+1}$. Let a be the vertex that b is matched to in $M_{>t}$. According to the definitions of the weights of edges in $R_{>t}$, we get that the weight of the edge $\langle b, b_0 \rangle \in E_t$ in $R_{>t}$ is

$$W_{>t}(\langle b_\ell, b_0 \rangle) = v_a(b) - v_a(b_0) \leq v_{>t}(b) \leq p(b), \tag{6}$$

where the first inequality is due to the definition of $v_{>t}(b)$, and the second inequality is due to our initial assumption. Combining (5) with (6) yields that the weight of the cycle $\langle b_0, b_1 \rangle, \langle b_1, b_2 \rangle, \ldots, \langle b_{\ell-1}, b_\ell \rangle, \langle b_\ell, b_0 \rangle$ in $R_{>t}$ is $\sum_{i=0}^{\ell} W_{>t}(\langle b_i, b_{i+1 \bmod \ell} \rangle) < \Delta$. Since Δ is the minimal weight of a positive cycle in $R_{>t}$, we get that either the weight of the cycle is negative, which contradicts Lemma 1, or the cycle is of weight 0, contradicting the fact the we delete every edge that takes part in some cycle of weight 0 in $R_{>t}$ from $R'_{>t}$. ■

4 No Optimal Dynamic Pricing Scheme for Coverage Valuations

We show an instance with agents with coverage valuations[3] for which no dynamic pricing scheme guarantees an optimal allocation. Interestingly, this instance admits Walrasian prices and has a unique optimal allocation, so no combination of these conditions is sufficient to imply optimal dynamic pricing schemes.

Theorem 4.1 *There exists an instance with agents with coverage valuations such that no dynamic pricing scheme guarantees more than a fraction $\frac{7.5}{8}$ of the optimal social welfare. This instance admits Walrasian prices.*

Proof: Let $I = \{a, b, c, d\}$ be a set of items and $N = \{1, 2, 3, 4\}$ be a set of agents. Agents 2, 3 and 4 are unit demand with the following valuation functions:

$$v_2(S) = \begin{cases} 2 & S \cap \{a, b\} \neq \emptyset \\ 0 & otherwise \end{cases}, v_3(S) = \begin{cases} 2 & S \cap \{a, c\} \neq \emptyset \\ 0 & otherwise \end{cases}, v_4(S) = \begin{cases} 1 & S \cap \{d\} \neq \emptyset \\ 0 & otherwise \end{cases}.$$

In addition, agent 1 has the following coverage valuation:

$$v_1(S) = \begin{cases} 2 & S = \{b\}, s = \{c\} \\ 3 & S = \{a\}, s = \{d\} \\ 3.5 & S = \{a, b\}, S = \{a, c\}, S = \{d, b\}, S = \{d, c\}, S = \{a, d\} \\ 3.75 & S = \{a, b, d\}, S = \{a, c, d\} \\ 4 & \{b, c\} \subseteq S \end{cases}.$$

Coverage valuation: To see that this is a coverage valuation, consider the following explicit representation. Let $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ be the set of elements, with weights $w(e_1) = w(e_5) = 5/4$ and $w(e_i) = 1/4$ for $i \neq 1, 5$. Item a covers the set $\{e_1, e_2, e_5, e_6\}$, item b covers the set $\{e_1, e_2, e_3, e_4\}$, item c covers the set $\{e_5, e_6, e_7, e_8\}$, and item d covers the set $\{e_1, e_4, e_5, e_8\}$.

Unique optimal allocation: The unique optimal allocation is to allocate item a to agent 1, item b to agent 2, item c to agent 3 and item d to agent 4. This allocation obtains social welfare of 8.

Walrasian prices: One can easily verify that the unique optimal allocation along with pricing each item at 1 is a Walrasian equilibrium.

We now show that no dynamic pricing scheme guarantees more than a fraction $\frac{7.5}{8}$ of the optimal allocation. In order to guarantee an optimal allocation, the following conditions must be satisfied:

- Agent 4's utility from item d should be strictly positive; i.e.,

$$p(d) < v_4(d) = 1. \tag{7}$$

- Agent 1 should strictly prefer item a over item d, i.e.,

$$v_1(a) - p(a) > v_1(d) - p(d) \Rightarrow p(a) < p(d). \tag{8}$$

- Agent 2 should strictly prefer item b over item a, i.e.,

$$v_2(b) - p(b) > v_2(a) - p(a) \Rightarrow p(b) < p(a). \tag{9}$$

[3]The class of coverage valuations is a strict subclass of submodular valuations.

- Agent 3 should strictly prefer item c over item a, i.e.,:

$$p(c) < p(a). \tag{10}$$

- Agent 1 should strictly prefer item a over the bundle $\{b, c\}$, i.e.,

$$v_1(a) - p(a) > v_1(\{b, c\}) - p(b) - p(c) \Rightarrow p(b) + p(c) - p(a) > 1. \tag{11}$$

Combining Equations (7) and (8) implies that $p(a) < 1$, while combining Equations (9), (10) and (11) yields $p(a) > 1$. Therefore, for every prices one might set, the adversary can set an order for which the first agent picks a different item than the one allocated to her in the optimal allocation.

Remark: note that the valuation function of agent 1 is not gross substitutes. In particular, her demand under prices $p(a) = p(c) = p(d) = 0$ and $p(b) = \epsilon$ is $\{b, c\}$, but if the price of item c increases to 2, then the unique bundle in the demand of agent 1 is $\{a, d\}$. ■

5 A 1/2-Approximate Static Pricing Scheme for any Class of Valuations

In this section we show that, given a partition of the items into bundles, pricing each bundle half of its value to the buyer guarantees half of the social welfare of the partition. Let $\mathcal{B} = \{B_1, B_2, \ldots, B_n\} \in \left(2^I\right)^n$ be a partition of the items such that $\bigcup_i B_i = I$ and for every $i \neq j$ $B_i \cap B_j = \emptyset$. Let $W = \sum_i v_i(B_i)$. We have the following:

Theorem 5.1 *Let $p : \mathcal{B} \rightarrow \Re_{\geq 0}$ be static bundle prices such that for every i, $p(B_i) = v_i(B_i)/2$. This pricing scheme achieves a welfare of at least $W/2$.*

Proof: Let $\mathbf{x}$ be an allocation which is a result of agents arriving at an arbitrary order, each taking their favorite bundles. Notice that the utility of an agent for acquiring the bundles in x_i is $u_i(x_i, P) = v_i(\bigcup_{B \in x_i} B) - \sum_{B \in x_i} p(B)$. Let $\mathbb{I}_i$ be an indicator which gets 1 if bundle B_i was acquired by some agent and 0 otherwise. Rearranging and summing over all the agents gives us:

$$\begin{aligned} \sum_i v_i\left(\bigcup_{B \in x_i} B\right) &= \sum_i \left(u_i(x_i, P) + \sum_{B \in x_i} p(B)\right) \\ &= \sum_i u_i(x_i, P) + \mathbb{I}_i p(B_i). \end{aligned} \tag{12}$$

We show that for every i, $u_i(x_i, P) + \mathbb{I}_i p(B_i) \geq v_i(B_i)/2$. Using (12) this is enough to prove the claim. For some i, either bundle B_i is purchased by some agent, in which case $\mathbb{I}_i p(B_i) = v_i(B_i)/2$. Otherwise, when agent i arrived, she could have purchased bundle B_i, for which she would have gotten a utility of $v_i(B_i) - p(B_i) = v_i(B_i)/2$. Since she bought the bundles which maximized her utility, her utility can only be greater than that, meaning $u_i(x_i, p) \geq v_i(B_i)/2$. ■

6 Optimal Pricing Schemes for Gross Substitutes Valuations for Unique Optimal Allocations

In Section 4 we have shown a case where there is a unique optimum, there exist Walrasian prices over the items, and no dynamic bundle pricing scheme can guarantee an optimal outcome. We first show that in case of a unique optimum, there is no need to search for a dynamic pricing scheme that retrieves the optimal allocation, since that the existence of such a scheme implies static prices that guarantee an optimal allocation.

Observation 6.1 *Let* $\mathbf{v} = (v_1, \ldots, v_n)$*, where* $v_i : 2^I \mapsto \Re_{\geq 0}$*, and let* $\langle \mathbf{v}, I \rangle$ *be an instance where* $\mathcal{B} = \{B_1, \ldots, B_n\}$ *is the unique partition of items that maximizes social welfare. If there exists an optimal dynamic bundle-pricing scheme, then there must exist an optimal static bundle-pricing scheme.*

Proof: Let $p_1 : \mathcal{B} \to \Re_{\geq 0}$ be the initial prices the optimal dynamic pricing scheme gives to the bundles. We claim that sticking to these prices throughout the process guarantees an optimal allocation as well. Without loss of generality, assume that agents with lower index arrive earlier and that the i-th agent to arrive is the first agent whose choice $X \neq \{B_i\}$ (could be that $X = \{B_j\}$, $j \neq i$, could be that $x = \{B_i, B_j, \ldots\}$, $j \neq i$, and could be that $X = \emptyset$).

It must be the case that $u_i(p_1, X) \geq u_i(p_1, B_i)$. Therefore, if this agent arrives first, she is not guaranteed to take $\{B_i\}$ since this not the unique bundle that maximizes her utility. This contradicts the optimality of the dynamic pricing scheme. ■

We previously showed that Walrasian prices do not imply the existence of optimal static prices in the case of a unique optimal allocation. However, for gross substitute valuations, the canonical valuation class for which Walrasian equilibrium is guaranteed to exist, if there exists a unique optimal allocation, optimal static prices do exist. We show how to compute such prices via a combinatorial algorithm inspired by Murota [Mur96a, Mur96b].[4]

Given some set of items $A \subseteq I$, we define the sets of item local to A as following $\textbf{Local}(A) = \{B \neq A \subseteq I : |B \setminus A| \leq 1 \text{ and } |A \setminus B| \leq 1\}$ We present the following alternative definition of gross substitute valuations [GS99]:

Definition 6.1 *A valuation* $v : 2^I \to \Re_{\geq 0}$ *satisfies the gross substitute condition if for every item prices* $p : I \to \Re_{\geq 0}$*, if there exists some* $A \subseteq I$ *such that* $A \notin D(I, p)$ *then there exists* $B \in \textbf{Local}(A)$ *such that* $u(B, p) > u(A, p)$*.*

We refer to this characterization as the *local improvement* property (LI).

Given a set of gross-substitute valuations and items $\langle \mathbf{v}, I \rangle$, let $\mathcal{B} = \{B_1, \ldots, B_n\}$ be the unique optimal allocation. We compute the prices $p : I \to \Re_{\geq 0}$ as follows:

1. Let $D = \{d_1, \ldots, d_n\}$ be a set of dummy items (one for each agent), $I' = I \cup D$ be the set of items after we added the dummy items. We extend every valuation function v_i to the domain $2^{I'}$, where $v_i(X) = v_i(X \cap M)$ (i.e., the dummy items have no effect on the value of the bundle). Define $\mathcal{B}' = \{B'_1, \ldots, B'_n\}$ where every bundle $B'_i = \{B_i \cup \{d_i\}\}$ receives an additional dummy item.

2. Let $R = \langle V = I', E \subset V \times V, W : E \to \Re \rangle$ (the exchange graph) be a weighted directed graph where:

    - $E = \{\langle a, b \rangle \in M^2 : a \in B'_i, b \in M \setminus B'_i \text{ for every } i\} \setminus D^2$: I.e., there is an edge from every item in some bundle B'_i to every item not in B'_i, *unless* the two items are dummy items.

[4] See [Lem14] for a concise description on how Murota's work relates to the computation of Walrasian prices for GS valuations.

- Let $e = \langle a, b\rangle$ where $a \in B'_i$ of some agent i be an edge in the graph. $W(e) = v_i(B'_i) - v_i(B'_i - a + b)$, i.e., the value of the agent from bundle B'_i *minus* the value she gets if she exchanges item a for item b.

3. Let $\delta > 0$ be the weight of minimum weight cycle in R (by Lemma 7, all the cycles in R are of strictly positive weight). Let $\gamma > 0$ be the weight of the minimum weight path out of all the paths from any vertex to any dummy vertex (all such paths are of strictly positive weight by Lemma 8). Let $\epsilon = \frac{\min\{\delta,\gamma\}}{n+1}$.

4. Update the weights by setting $W(e) \leftarrow W(e) - \epsilon$ for every edge e in the graph.

5. Price the items using algorithm Price-Items (Figure 2) with graph R as input.

Lemma 7 *All the cycles in the graph R described in step 2 of the above price computation are of strictly positive weight.*

Proof: Let i be some agent (recall that B_i the bundle allocated to her in the unique optimal allocation). Let $\delta = \min_{\mathbf{x}\neq\mathcal{B}}\{\mathsf{SW}(\mathcal{B},\mathbf{v}) - \mathsf{SW}(\mathbf{x},\mathbf{v})\}$ be the difference in welfare between the optimal allocation, and the second best allocation. $\delta > 0$ since the optimal allocation is unique. For some item $b \in I \setminus B_i$ define the modified valuation $v_i^{(b)} : 2^{I'} \to \Re_{\geq 0}$ as follows:

$$v_i^{(b)}(S) = \begin{cases} v_i(S) + \delta & b \in S \\ v_i(S) & b \notin S \end{cases}. \tag{13}$$

Let $\mathbf{v}^{(b)} = (v_{-i}, v_i^{(b)})$. For some arbitrary allocation $\mathbf{x} \neq \mathcal{B}$ we have

$$\begin{aligned} \mathsf{SW}(\mathbf{x},\mathbf{v}^{(b)}) &= v_i^{(b)}(x_i) + \sum_{j\neq i} v_j(x_j) \\ &\leq v_i(x_i) + \delta + \sum_{j\neq i} v_j(x_j) \\ &\leq \mathsf{SW}(\mathcal{B},\mathbf{v}) \\ &= \mathsf{SW}(\mathcal{B},\mathbf{v}^{(b)}), \end{aligned}$$

and therefore, $\mathcal{B}$ is still an optimal allocation for profile $\mathbf{v}^{(b)}$. We next claim that $v_i^{(b)}$ is gross substitute. We use the characterization of Reijnierse et al. [RvGP02]:

Definition 6.2 *A valuation $v : 2^I \to \Re_{\geq 0}$ is gross substitute if and only if v is submodular, and for every $S \subset I$ and $b_1, b_2, b_3 \notin S$:*

$$v(S \cup \{b_1, b_2\}) + v(S \cup \{b_3\}) \leq \max\{v(S \cup \{b_1\}) + v(S \cup \{b_2, b_3\}), v(S \cup \{b_2\}) + v(S \cup \{b_1, b_3\})\}. \tag{14}$$

First we show that $v_i^{(b)}$ is submodular. Let $S \subset T$ two sets of items, and let b' be some item. if $b' \neq b$, then we know that $v_i^{(b)}(b'|S) = v_i(b'|S) \leq v_i(b'|T) = v_i^{(b)}(b'|T)$. Otherwise, $v_i^{(b)}(b'|S) = v_i(b'|S) + \delta \leq v_i(b'|T) + \delta = v_i^{(b)}(b'|T)$. Next, we verify (14). Let S be some set of items and b_1, b_2, b_3 some items not in S. Since, v_i is GS, we know that (14) holds. Without loss of generality, let us assume that $v_i(S\cup\{b_1, b_2\}) + v_i(S\cup\{b_3\}) \leq v_i(S\cup\{b_1\}) + v_i(S\cup\{b_2, b_3\})$, which is equivalent to $v_i(b_2|S\cup\{b_1\}) \leq$

$v_i(b_2|S \cup \{b_3\})$. If $b_2 \neq b$ then $v_i^{(b)}(b_2|S \cup \{b_1\}) = v_i(b_2|S \cup \{b_1\}) \leq v_i(b_2|S \cup \{b_3\}) = v_i^{(b)}(b_2|S \cup \{b_3\})$, and otherwise $v_i^{(b)}(b_2|S \cup \{b_1\}) = v_i(b_2|S \cup \{b_1\}) + \delta \leq v_i(b_2|S \cup \{b_3\}) + \delta = v_i^{(b)}(b_2|S \cup \{b_3\})$. This implies that $v_i^{(b)}(S \cup \{b_1, b_2\}) + v_i^{(b)}(S \cup \{b_3\}) \leq v_i^{(b)}(S \cup \{b_1\}) + v_i^{(b)}(S \cup \{b_2, b_3\})$.

Since $\mathbf{v}^{(b)}$ is a gross substitute valuation profile, it admits a Walrasian equilibrium $(\mathcal{B}', p)$. We claim that $(\mathcal{B}', p)$ is also a Walrasian equilibrium for $\mathbf{v}$. This is true since $v_i(B'_i) = v_i^{(b)}(B'_i)$, and for every S, $v_i(S) \leq v_i^{(b)}(S)$.

For some item $b' \in I'$, we denote by $N(b')$ the function that returns the agent j for which $b' \in B'_j$. Consider a cycle in R that uses edge $\langle a, b \rangle$ for some cycle in R. Let $(b_0, b_1, \ldots, b_{\ell-1}, b_0)$ denote the cycle, where $b_0 = a$ and $b_1 = b$. We denote $b_\ell = b_0$. Since $(\mathcal{B}', p)$ is a Walrasian equilibrium for $\mathbf{v}^{(b)}$, we know that

$$\begin{aligned}
v_i(B'_i) - p(B'_i) &= v_i^{(b)}(B'_i) - p(B'_i) \\
&\geq v_i^{(b)}(B'_i - a + b) - p(B'_i - a + b) \\
&= v_i(B'_i - a + b) + \delta - p(B'_i - a + b) \\
&> v_i(B'_i - a + b) - p(B'_i - a + b).
\end{aligned}$$

Rearranging gives us

$$\begin{aligned}
W(\langle b_0, b_1 \rangle) &= W(\langle a, b \rangle) \\
&= v_i(B'_i) - v_i(B'_i - a + b) \\
&> p(B'_i) - p(B'_i - a + b) \\
&= p(a) - p(b) \\
&= p(b_0) - p(b_1). \qquad (15)
\end{aligned}$$

Since $(\mathcal{B}', p)$ is a Walrasian equilibrium for $\mathbf{v}$ as well, we get that for every $j \in 1, \ldots, \ell - 1$,

$$v_{N(b_j)}(B'_{N(b_j)}) - p(B'_{N(b_j)}) \geq v_{N(b_j)}(B'_{N(b_j)} - b_j + b_{j+1}) - p(B'_{N(b_j)} - b_j + b_{j+1}).$$

Rearranging gives us

$$\begin{aligned}
W(\langle b_j, b_{j+1} \rangle) &= v_{N(b_j)}(B'_{N(b_j)}) - v_{N(b_j)}(B'_{N(b_j)} - b_j + b_{j+1}) \\
&\geq p(b_j) - p(b_{j+1}). \qquad (16)
\end{aligned}$$

Summing inequality (15) with inequalities of type (16) for all $j \in 1, \ldots, \ell - 1$ gives us that the weight of the cycle $(b_0, b_1, \ldots, b_{\ell-1}, b_0)$ is

$$\sum_{j \in \{0, \ldots, \ell-1\}} W(\langle b_j, b_{j+1} \rangle) > \sum_{j \in \{0, \ldots, \ell-1\}} (p(b_j) - p(b_{j+1})) = 0.$$

Since agent i is an arbitrary agent and item b is an arbitrary (non-dummy) item, we get that all the cycles in R that use an edge which ends in a non-dummy item must be strictly positive. Since there are no edges who between two dummy items in R, we get that all cycles must use at least one edge which ends in a non-dummy item, hence, must be strictly positive. ■

We now show a property which is crucial in establishing that the price of every dummy node is zero.

Lemma 8 *Let R be the graph described in step 2 of the above price computation. For every agent i, dummy node d_i and every item $b \in I' \setminus \{d_i\}$, $\text{dist}_R(b, d_i) > 0$.*

Proof: Let d_i be a dummy item added to the bundle of some agent i. Let b be some item in $I' \setminus \{d_i\}$. For some dummy item $d_j \neq d_i$, let R_{d_i,d_j} be the graph established by taking graph R (after step 2), and adding an edge $\langle d_i, d_j \rangle$ of weight $W(\langle d_i, d_j \rangle) = V(B'_i) - V(B'_i - d_i + d_j) = 0$. First notice using a similar argument to the one presented in the proof of Lemma 7, it is not hard to see that all the cycles in the graph R_{d_i,d_j} are of strictly positive weight for any choice of d_j. We use $b \rightsquigarrow d_i$ and $W(b \rightsquigarrow d_i)$ to denote some simple path from b to d_i and its weight. We now consider the following cases:

- b is in $I \setminus B_i$: In this case, consider the cycle obtained by adding edge $\langle d_i, b \rangle$ to $b \rightsquigarrow d_i$. Since every cycle in R is of strictly positive weight, we have that $W(b \rightsquigarrow d_i) + W(\langle d_i, b \rangle) > 0$. Since $W(\langle d_i, b \rangle) = v_i(B_i) - v_i(B_i + b) \leq 0$, it must be the case where $W(b \rightsquigarrow d_i) > 0$.

- b is some dummy item $d_j \neq d_i$: Consider the graph R_{d_i,d_j} and the cycle obtained by adding edge $\langle d_i, d_j \rangle$ to $d_j \rightsquigarrow d_i$. Since every cycle in R_{d_i,d_j} is of strictly positive weight, we have $W(d_j \rightsquigarrow d_i) + W(\langle d_i, d_j \rangle) = W(d_j \rightsquigarrow d_i) > 0$.

- $b \in B_i$: Consider the graph R_{d_i,d_j}. Consider the cycle obtained by adding edges $\langle d_i, d_j \rangle, \langle d_j, b \rangle$ to $d_j \rightsquigarrow d_i$. We have that the weight of the cycle is

  $$W(b \rightsquigarrow d_i) + W(\langle d_i, d_j \rangle) + W(\langle d_j, b \rangle) = W(d_j \rightsquigarrow d_i) + W(\langle d_j, b \rangle) > 0.$$

  Since $W(\langle d_j, b \rangle) = v_j(B_j) - v_j(B_j + b) \leq 0$, we get $W(b \rightsquigarrow d_i) > 0$.

Since $W(b \rightsquigarrow d_i) > 0$ for every simple path from b to d_i, and there are no negative cycles in R, we have that $\text{dist}_R(b, d_i) > 0$.

■

From Lemmas 7 and 8 and by carefully choosing ϵ in step 4, we immediately get:

Corollary 6.2 ***After*** *updating the edge weights (step 4) all the cycles in R are of* ***strictly*** *positive weight, all the paths ending in a dummy vertex are of a* ***strictly*** *positive weight.*

It is crucial that we have the following:

Corollary 6.3 *For every dummy item d_i, $p(d_i) = 0$.*

Proof: By the way Price-Items operates, an item d_i has a price greater than 0 only if there exists a path of negative weight from some vertex to d_i. By Corollary 6.2, this cannot happen. ■

The next lemma shows that every for "local" change an agent may perform, her utility decreases.

Lemma 9 *For every agent i, for every bundle $C \in$* ***Local***(B_i)*, we have $u(p, B_i) > u(p, C)$.*

Proof: Let C be some bundle in **Local**(B_i). We inspect the following cases:

- $A \setminus C = \{a\}$ and $C \setminus A = \{b\}$: In this case, there is a directed edge in $\langle a, b \rangle \in E$ of weight $v_i(B_i) - v_i(B_i - a + b) - \epsilon = v_i(B_i) - v_i(C) - \epsilon$. By Lemma 3, $p(C) - p(B_i) \geq -v_i(B_i) + v_i(C) + \epsilon > v_i(C) - v_i(B_i)$. Rearranging gives us $u(p, B_i) = v_i(B_i) - p(B_i) > v_i(C) - p(C) = u(p, B_i)$.

- $A \setminus C = \{a\}$ and $C \setminus A = \emptyset$: There is an edge between a and some dummy item d_j of weight $v_i(B_i) - v_i(B_i - a + d_j) - \epsilon = v_i(B_i) - v_i(B_i - a) - \epsilon = v_i(B_i) - v_i(C) - \epsilon$. Again, by Lemma 3 we get that $u(p, B_i) > u(p, B_i)$.

- $A \setminus C = \emptyset$ and $C \setminus A = \{b\}$: There is an edge between d_i and b of weight $v_i(B_i) - v_i(B_i - d_i + b) - \epsilon = v_i(B_i) - v_i(B_i + b) - \epsilon = v_i(B_i) - v_i(C) - \epsilon$. Again, by Lemma 3 we get that $u(p, B_i) > u(p, B_i)$.

■

The following property of gross substitute valuations shows that the above lemma is enough to show that the prices achieve optimal social welfare.

Lemma 10 *Let $v : I \to \Re_{\geq 0}$ be a valuation that satisfies the gross substitute property, let $P : I \to \Re_{\geq 0}$ be some item pricing and let A be some set of items in $D(I, p)$. If $|D(I, p)| > 1$ then there exists some $B \in$ **Local**(A) such that $B \in D(I, p)$.*

Proof: Let A be some set in $D(I, p)$ and let us assume that $|D(I, p)| > 1$ and $D(I, p) \cap \textbf{Local}(A) = \emptyset$. Let us define the following set:

$$\textbf{Local}^{+}(p, A) = \{B \in \textbf{Local}(A) : \exists C \neq A \in D(I, p) \text{ s.t. } |B \Delta C| \leq |A \Delta C|\},$$

that is, the set of local sets to A that are more similar to another set in $D(I, p)$ than A is. Since $|D(I, p)| > 1$, $\textbf{Local}^{+}(p, A)$ is non empty. Let $B = \arg\min_{X \in \textbf{Local}^{+}(p,A)} \{u(p, A) - u(p, X)\}$, let $C \neq A$ be the set in $D(I, p)$ such that $|B \Delta C| \leq |A \Delta C|$ and let $\delta = \min_{X \in \textbf{Local}(A)} \{u(p, A) - u(p, X)\}$. $\delta > 0$ by our assumption. We define the following item pricing p':

- If $|B \setminus A| = 1$, then for $a \in B \setminus A$ set $p'(a) = p(a) - \delta/2$ and $p'(b) = p(b)$ for all other $b \in I - a$.

- Otherwise, let a be an item in $A \setminus B$. Set $p'(a) = p(a) + \delta/2$ and $p'(b) = p(b)$ for all other $b \in I - a$.

Notice that $C \in D(I, p')$, $A \notin D(I, p')$ and $D(I, p') \subset D(I, p)$. Therefore, $\textbf{Local}^{+}(p', A) \subseteq \textbf{Local}^{+}(p, A)$. If $|B \setminus A| = 1$ then for every set $X \in \textbf{Local}(A)$, we have that $u(p', A) = u(p, A) \geq u(p, X) + \delta > u(p', X)$. Otherwise, for every $X \in \textbf{Local}(A)$, $u(p', A) = u(p, A) - \delta/2 \geq u(p, X) + \delta/2 > u(p', X)$. Therefore, $A \notin \mathrm{D}(I, p)$, and there is no local improvement, contradicting the LI property of gross substitute valuations. ■

Theorem 6.4 *Item prices p computed above achieve an optimal welfare.*

Proof: By Lemma 9 for every agent i $u_i(p, B_i) > u_i(p, X)$ for every $X \in \textbf{Local}(B_i)$. By the LI property of v_i, we have that $B_i \in D_i(I, p)$. By Lemma 10 we get that $D_i(I, p) = \{B_i\}$. ■

7 An Optimal Static Bundle Pricing Scheme for Super-additive Valuations

A valuation $v : 2^I \to \Re_{\geq 0}$ is said to be super-additive if for every two disjoint sets of items $A, B \subseteq I$, $v(A \cup B) \geq v(A) + v(B)$. We show that in the case where all agents have super-additive valuations, it is possible to come up with bundles and bundle-prices such that for every arrival order of the agents, the resulting welfare is optimal. Let $\mathcal{B} = \{B_1, B_2, \ldots, B_n\} \in \left(2^I\right)^n$ be a partition of the items such that $\bigcup_i B_i = B$ and for every $i \neq j$ $B_i \cap B_j = \emptyset$. Let $W = \sum_i v_i(B_i)$. We present an algorithm that given a bundling $\mathcal{B}$, computes a bundling of the items and appropriate prices that guarantee a welfare of at least W.

Given bundling $\mathcal{B}' = \{B'_1, B'_2, \ldots, B'_n\}$ and a pricing over bundles $p : \mathcal{B}' \leftarrow \Re_{\geq 0}$, we denote utility of agent i for a collection of bundles $x \in 2^{B'}$ by $u_i(\mathcal{B}', p, x) = v_i\left(\bigcup_{B' \in x} B'\right) - \sum_{B' \in x} p(B')$. We assume we have access to demand oracles of the given agents, i.e., we can find for each agent i a collection of bundles $x_i \in \operatorname{argmax}_{x \in 2^{\mathcal{B}'}}\{u_i(\mathcal{B}', p, x)\}$. We denote the function that returns such a set of a maximal size by $D_i(\mathcal{B}', p)$ (meaning there is no $x' \supset x_i$ such that $x' \in \operatorname{argmax}_x\{u_i(\mathcal{B}', p, x)\}$). We also assume that given some bundle $B' \in \operatorname{argmax}_x\{u_i(x)\}$, we can find $\min\limits_{x \in 2^{\mathcal{B}'} : x \ni \tilde{B} \neq B'} \{u_i(\mathcal{B}', p, \{B'\}) - u_i(\mathcal{B}', p, x)\}$. We denote the function which computes this by $\text{mindiff}_i(\text{B}', \mathcal{B}', p)$.

Price-Super-Additive

Input: Additive valuations $v_1, \ldots, v_n$, initial bundling $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ with welfare W.

Output: Bundling $\mathcal{B}' = \{B'_1, B'_2, \ldots, B'_n\}$ and pricing $p : \mathcal{B}' \rightarrow \Re_{\geq 0}$ with welfare guarantee $\geq W$.

1. Initialize $B'_i \leftarrow B_i$ and $p(B'_i) \leftarrow v_i(B'_i)$ for every i.
2. While $\exists B' \neq B'_i \in D_i(\mathcal{B}', p)$ for some agent i:
   (a) $x_i \leftarrow D_i(\mathcal{B}', p)$; $B'_i \leftarrow \bigcup\limits_{B' \in x_i} B'$.
   (b) $\forall j \neq i$ such that $B'_j \in x_i$, $B'_j \leftarrow \emptyset$.
   (c) $p(B'_j) = v_j(B'_j)$ for every agent j.
3. Let $\delta \leftarrow \min_i \text{mindiff}_i(B'_i, \mathcal{B}', p)$; $\epsilon = \frac{\delta}{n}$.
4. For every i, set $p(B'_i) \leftarrow p(B'_i) - \epsilon$.

Figure 3: Computing bundle prices for super-additive valuations.

The following lemma shows that all the items remain allocated at all time.

Lemma 11 *For every x_i selected in step 2a, we have that $B'_i \in x_i$.*

Proof: Assume not. Then

$$\begin{aligned} u_i(\mathcal{B}', p, x_i \cup \{B'_i\}) &= v_i\left(B'_i \cup \bigcup_{B' \in x_i} B'\right) - p(B'_i) - \sum_{B' \in x_i} p(B') \\ &\geq v_i(B'_i) - p(B'_i) + v_i\left(\bigcup_{B' \in x_i} B'\right) - \sum_{B' \in x_i} p(B') \\ &= u_i(\mathcal{B}', p, x_i), \end{aligned}$$

where the inequality follows from super-additivity. This contradicts the maximality of x_i. ■

The following lemma shows that the resulting bundling $\mathcal{B}'$ has a welfare at least as high as the initial bundling $\mathcal{B}$.

Lemma 12 *Let $B' = \{B'_1, \ldots, B'_n\}$ be the bundling computed by* Price-Super-Additive*, we have that $\sum_i v_i(B'_i) \geq W$.*

Proof:

Let $\mathcal{B}'_1 = \{B'^1_1, \ldots B'^1_n\}$ and $\mathcal{B}'_2 = \{B'^2_1, \ldots B'^2_n\}$, and p_1, p_2 be the bundles and pricing functions before and after some iteration. Let $A_\emptyset = \{j : B'^1_j \neq \emptyset \wedge B'^2_j = \emptyset\}$. Since agent i received a set of bundles in his demand, it must be the case that $v_1\left(\bigcup_{j \in A_\emptyset} B'^1_j | B'^1_i\right) \geq \sum_{j \in A_\emptyset} p_1(B'^1_j) = \sum_{j \in A_\emptyset} v_j(B'^1_j)$. Therefore, $v_i\left(B'^2_i\right) = v_i\left(B'^1_i \cup \bigcup_{j \in A_\emptyset} B'^1_j\right) = v_i\left(B'^1_i\right) + v_i\left(\bigcup_{j \in A_\emptyset} B'^1_j | B'^1_i\right) \geq v_i\left(B'^1_i\right) + \sum_{j \in A_\emptyset} v_j(B'^1_j)$. We get that the value of agent i for her bundle after the iteration is greater than her value before the iteration *plus* the sum of values of all the agents whose bundle has changed during the iteration. Clearly, $\sum_j v_j(B'^2_j) \geq \sum_j v_j(B'^1_j)$. Since this holds after every iteration, it must be that the welfare out of the final bundling is at least the welfare out of the initial bundling. ■

Finally, we show that every agent i strictly prefers B'_i to every other outcome.

Lemma 13 *Let $\mathcal{B}'$ and p be the bundling and price over bundles computed by* Price-Super-Additive*. For every agent i, $\{B'_i\}$ is the only item in $\operatorname{argmax}_x\{u_i(\mathcal{B}', p, x)\}$.*

Proof: First, note that $u_i(\mathcal{B}', p, B'_i) = \epsilon > 0$. Let $p' : \mathcal{B}' \to \Re_{\geq 0}$ be a pricing over bundles such that $p'(B'_i) = v_i(B'_i)$ for every i (i.e., the prices before they were changed on the last step of Price-Super-Additive). Let $x \neq \{B'_i\}$ be some non-empty set of bundles in $2^{\mathcal{B}'}$. By the definition of δ, we have that $u_i(\mathcal{B}', p', x) = u_i(\mathcal{B}', p', x) - u_i(\mathcal{B}', p', \{B'_i\}) \leq -\delta$. Therefore, for pricing function p we have that $u_i(\mathcal{B}', p, x) \leq u_i(\mathcal{B}', p', x) + n\epsilon \leq -\delta + \delta = 0$. ■

We get the following immediate corollary.

Corollary 7.1 *In case the agents have super-additive valuations, there exists bundling and static prices over bundles such that for any arrival order of the agents, the resulting allocation is optimal.*

Proof: Given a SW maximizing bundling $\mathcal{B}$, Price-Super-Additive returns an optimal bundling $\mathcal{B}'$ (Lemma 12). Moreover, bundle price p are such that every agent arriving agent i takes exactly bundle B'_i (Lemma 13). ■

8 Figures

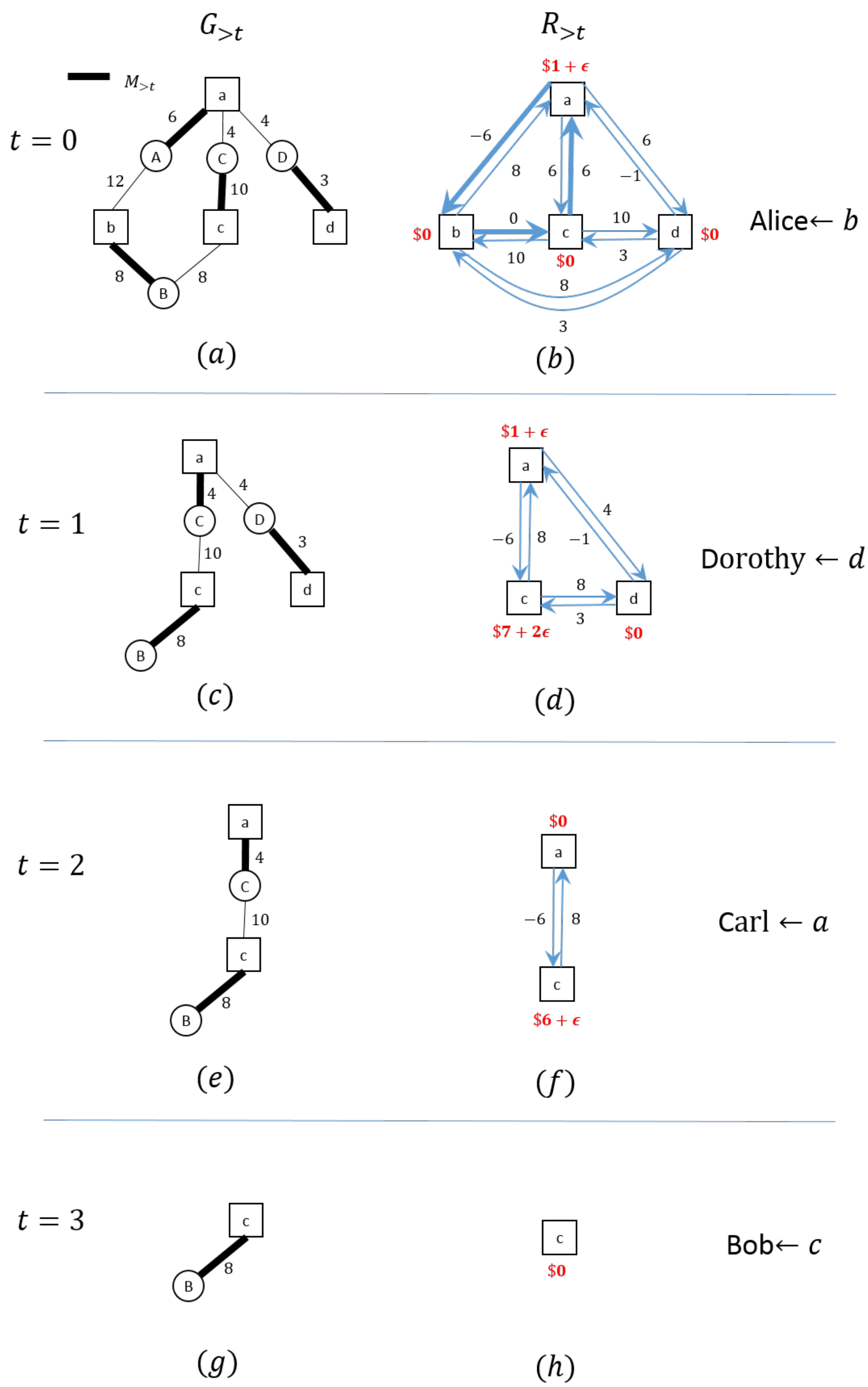

Figure 4: Phases $t = 0, 1, 2, 3$ of our running example. Squares represent items and circles represent buyers. Every row represent a phase in the process, where a single buyer arrives. On the left one sees the graph representing the valuations of the remaining buyers and items (grap;hs labeled (a), (b), (c) and (d), where thick edges represent a maximal matching in the graph. Graphs labeled (b), (d), (f) and (h) give the graphs $R_{>t}$ from which the dynamic are computed. Directed cycles of length 0 (if any) are represented by thick edges, after they are discarded prices are computed via Algorithm Price-Items. On the very right one sees the next buyer to arrive as well the item they choose (based upon the pricing, and breaking ties for the sake of this example.

References

[BH08] Liad Blumrosen and Thomas Holenstein. Posted prices vs. negotiations: an asymptotic analysis. In *EC*, page 49, 2008.

[CEFJ15] Ilan Reuven Cohen, Alon Eden, Amos Fiat, and Łukasz Jeż. Pricing online decisions: Beyond auctions. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on discrete algorithms*, pages 73–91. SIAM, 2015.

[CHK07] Shuchi Chawla, Jason D. Hartline, and Robert D. Kleinberg. Algorithmic pricing via virtual valuations. In *ACM Conference on Electronic Commerce*, pages 243–251, 2007.

[CHMS10] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC*, pages 311–320, 2010.

[CMS10] Shuchi Chawla, David L. Malec, and Balasubramanian Sivan. The Power of Randomness in Bayesian Optimal Mechanism Design. In *the 11th ACM Conference on Electronic Commerce (EC)*, 2010.

[FGL13] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial walrasian equilibrium. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 61–70. ACM, 2013.

[FGL15] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 123–135. SIAM, 2015.

[GS99] Faruk Gul and Ennio Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.

[KJC82] Alexander S Kelso Jr and Vincent P Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society*, pages 1483–1504, 1982.

[KVV90] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[Lem14] R Paes Leme. Gross substitutability: An algorithmic survey. *preprint*, 2014.

[Leo83] Herman B Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–79, 1983.

[Mur96a] Kazuo Murota. Valuated matroid intersection i: Optimality criteria. *SIAM Journal on Discrete Mathematics*, 9(4):545–561, 1996.

[Mur96b] Kazuo Murota. Valuated matroid intersection ii: Algorithms. *SIAM Journal on Discrete Mathematics*, 9(4):562–576, 1996.

[NS06] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129:192–224, 2006.

[RvGP02] Hans Reijnierse, Anita van Gellekom, and Jos AM Potters. Verifying gross substitutability. *Economic Theory*, 20(4):767–776, 2002.

[Wal74] L. Walras. *Éléments d'économie politique pure; ou, Théorie de la richesse sociale.* Corbaz, 1874.

A Optimal Social Welfare for k-Demand Item-Dependent Valuations

Let $G = (N \cup I, E, \mathbf{v})$ be a bipartite graph. We say that $v : 2^I \to \Re_{\geq 0}$ is k-demand if there exists some bound k on the number of items an agent can benefit from — the agent's value out of getting bundle B is

$$v(B) = \max_{X \subseteq B : |X| \leq k} \sum_{b \in X} v(b).$$

We say that a valuation profile $\mathbf{v} = \{v_1, \ldots, v_n\}$ is *item-dependent* if there exists some function $w : I \to \Re_{\geq 0}$ such that for every agent i and every item b, $v_i(b) \in \{0, w(b)\}$. Finally, we say that a valuation profile is $\mathbf{k}$-*demand item-dependent* for some vector $\mathbf{k} = (k_1, \ldots, k_n)$ if $\mathbf{v}$ is item-dependent and for every i v_i is k_i-demand. Our main result of this section is

Theorem A.1 *For any vector* $\mathbf{k} = (k_1, \ldots, k_n)$ *and for every valuation profile which is* $\mathbf{k}$-*demand item-dependent there exists an optimal dynamic bundle-pricing scheme.*

We say that a partition $\mathcal{B}_0$ of goods into bundles *respects* another partition $\mathcal{B}_1$ of good into bundles if for any two items u, v that belong to a bundle $B_0 \in \mathcal{B}_0$, there exists a bundle $B_1 \in \mathcal{B}_1$ that contains both u and v.

Given an allocation of bundles to the agents, we define the *relation* directed graph $R = (V, E)$ as follows. Let B_i be the set of items assigned to agent i. For each such bundle B_i, V contains a vertex s_i. There is an edge from vertex s_i to vertex s_j in E if $v_j(B_j) = v_i(B_j)$. Note that in any optimal allocation, there is no clients i, j such that $v_j(B_j) < v_i(B_j)$.

Let $\epsilon < \min_{u \in I} v(u)$. The algorithm at time 0, starts with a bundle for each good.
For each time t,

Price-**k**-Demand

Input: A set of bundles $\mathcal{B}_{t-1}$ and a set of agents N_{t-1} and the valuation $v_i : \mathcal{B}_{t-1} \to \Re_{\geq 0}$ for each agent i.

Output: A set of bundles $\mathcal{B}_t$ which respects $\mathcal{B}_{t-1}$ and an assignment of prices p_t to the bundles of B_t.

1. Compute an optimal allocation x_t of the bundles of $\mathcal{B}_t$ to the agents of N_{t-1}.
2. For each set $\mathcal{B}(x_t, i)$ of bundles assigned to agent i in x_t, create a bundle $B_i^t \leftarrow \mathcal{B}(x_t, i)$.
3. Construct the relation graph R_t from the bundles and greedily remove cycles to obtain a DAG, DAG_t: Remove each edge of R_t that takes part in at least one directed cycle of R_t.
4. Apply a topological sort to D. It defines an ordering σ of the bundles.
5. For any bundle of rank r in σ, define $p_t(B_i^t) \leftarrow v(B_i) - \epsilon^r$.
6. Return $\{B_0^t, \ldots, B_n^t\}$ and p_t

Figure 5: A pricing algorithm for the Vertex Weighted k_i Unit Demand case.

We first prove several invariants of Procedure 5.

Lemma A.2 *For any time t, for any $s_i, s_j \in DAG_t$, if there is an edge $< s_i, s_t > \in DAG_t$ then $u_i(B_i) > u_i(B_j)$ for the pricing p_t.*

Proof: Suppose that s_i is at rank r in σ and s_j at rank $r' > r$. We have that $u_i(B_i) = v_i(B_i) - v(B_i) + \epsilon^r$ and $u_i(B_j) = v_i(B_j) - v(B_j) + \epsilon^{r'}$. Since there is an edge $< s_i, s_t > \in \mathrm{DAG}_t$, we have $v_i(B_j) = v_j(B_j) = v(B_j)$. Moreover, $v_i(B_i) = v(B_i)$. It follows that $u_i(B_i) = \epsilon^r$ and $u_i(B_j) = \epsilon^{r'}$. Hence, $u_i(B_i) > u_i(B_j)$. ■

Lemma A.3 *At any time t, the arriving agent i only picks bundles whose corresponding vertices s_j have ingoing edges from s_i in R_t or B_i.*

Proof: Assume toward contradiction that this is not the case and that agent i picks a bundle B_j whose corresponding vertex s_j has no ingoing edge from s_i in R_t. If there is no edge from s_i to s_j then $v_j(B_j) \geq v_i(B_j) + \min_{u \in I} v(u)$. Moreover, $p_t(B_j) \geq v(B_j) - \epsilon$. It follows that $u_i(B_j) = v_i(B_j) - p_t(B_j)$. Hence, $u_i(B_j) \leq v_i(B_j) - v_j(B_j) + \epsilon$. Therefore, by the choice of ϵ, $u_i(B_j) < 0$, a contradiction.

■

We now proceed to the proof of Theorem A.1.

Proof: [Proof of Theorem A.1] Consider a time t and let $\{a_0, \ldots, a_{t-1}\}$ be the set of agents that arrived at times $0, \ldots, t-1$. Moreover, let $B(a_j)$ be the set of bundles that agent $a_j \in \{a_0, \ldots, a_{t-1}\}$ bought when he arrived at time j and let x_t^{Alg} denote the allocation of the bundles definied by $\{(a_0, B(a_0)), \ldots, (a_{t-1}, B(a_{t-1}))\}$. We aim at proving the following invariant.
At any time t, there exists an optimal allocation OPT_t which respects x_t^{Alg}.
This is trivially true for $t = 0$ and we show by induction that it remains true for any time $t > 0$. Consider a time t and the agent i arriving at time t. We show that there exists an allocation that assigns to agent i the same bundles that agent i buys at time t and which achieves an optimal social welfare. Let x denote an allocation of the bundles of optimal social welfare before i arrives. Such an allocation is guaranteed to exist by induction hypothesis. Let $\mathcal{B}_i$ denote the set of bundles that agent i picks when it arrives.

We consider the set $\beta \subseteq \mathcal{B}_i$ which contains the bundles whose corresponding vertices have an ingoing edge from s_i in R_t. We define a new allocation x^*, where each bundle of β is assigned to agent i in addition to bundle B_i and any other agent j is assigned $B_j \setminus \beta$. We argue that x^* achieves a social welfare value of at least the social welfare value of x. Recall that for any bundle $B_j \in \beta$, $v_i(B_j) - v(B_j) \geq 0$. We now compare the cost of x^* and x: the difference of value received by agent i is

$$\sum_{B_j \in \beta} v_i(B_j) \geq \sum_{B_j \in \beta} v(B_j),$$

and for any agent j such that B_j is in β, we have

$$-v_j(B_j) \geq -v(B_j).$$

Combining and summing over all each client j such that $B_j \in \beta$, we obtain that the social welfare of allocation x^* is at least the social welfare value of x. Hence, if B_i is included in $\mathcal{B}_i$, then by Lemma A.3, we are done.

Now, suppose B_i is not in $\mathcal{B}_i$. Then, there exists a bundle B_j whose corresponding vertex s_j has an ingoing edge in R_t from s_i which does not appear in DAG_t, i.e: which was removed at step 3 of Procedure 5, as otherwise, B_i would lead to a better utility for agent i, by Lemma A.2.

For such a bundle B_j, there exists a directed cycle C which contains the edge $< s_i, s_j >$ in R_t. We now consider the allocation $\hat{x}$ where agent i receives the same bundles than in x^* minus bundle B_i. Moreover for each edge $< s_j, s_k > \in C$, bundle B_k is assigned to agent j instead of agent k. For the other agent, the assignment is the same as in x^*.

We argue that this allocation achieves a social welfare value of at least the social welfare value of x^*. Indeed for any edge $< s_k, s_\ell > \in C$, we have that $v_k(B_\ell) \geq v(B_\ell) \geq v_\ell(B_\ell)$. By summing over all k such that s_k is a vertex of C, we obtain that the total difference between the the social welfare values of allocations $\hat{x}$ and x^* is greater or equal to 0. Therefore, the induction hypothesis is met and the Theorem follows. ■

B No Static Prices for the Running Example (Figure 4)

Lemma B.1 *There is no static pricing scheme for the running example that achieves optimal welfare.*

Proof: Note that in any welfare maximizing allocation for the example, all items should be allocated. We consider pricing Let D_{Alice}, D_{Bob} and, D_{Carl} denote the demand sets of Alice, Bob, and Carl, respectively, under pricing p.

Suppose that $|D_{\text{Alice}}| = 2$. Then if $c \notin D_{\text{Bob}} \cup D_{\text{Carl}}$, then $D_{\text{Bob}} = \{b\}$ and $D_{\text{Carl}} = \{a\}$. We consider the following sequence: Carl arrives first and takes a, Bob arrives second and takes b and so c is not picked, a contradiction. So suppose c belongs to D_{Bob}, then we consider the following order of arrival: Bob arrives first and takes c, Alice arrives second and takes a and so b is not picked, a contradiction. Similarly if $c \in D_{\text{Carl}}$, we consider the arrival where Carl arrives first and takes c, Alice arrives second and takes b and so at least one of a or d is not picked, a contradiction. Symmetrically, the above argument applies to the cases where $|D_{\text{Bob}}| = 2$ or $|D_{\text{Carl}}| = 2$.

Then, suppose that $|D_{\text{Alice}}|, |D_{\text{Bob}}|, |D_{\text{Carl}}| = 1$. Suppose first that $D_{\text{Alice}} = \{a\}$ and so, $D_{\text{Bob}} = \{b\}$ and $D_{\text{Carl}} = \{c\}$. Then $6 - p(a) > 12 - p(b)$, $8 - p(b) > 8 - p(c)$ and $10 - p(c) > 4 - p(a)$. Combining we obtain, $6 + p(a) > p(c) > p(b) > 6 + p(a)$ a contradiction. Suppose then that $D_{\text{Alice}} = \{b\}$ and so, $D_{\text{Bob}} = \{c\}$ and $D_{\text{Carl}} = \{d\}$. Then $12 - p(b) > 6 - p(a)$, $8 - p(c) > 8 - p(b)$ and $4 - p(a) > 10 - p(c)$. Combining, $6 + p(a) > p(b) > p(c) > 6 + p(a)$, a contradiction. The assertion of the lemma follows. ■