https://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 49, No. 3, August 2024, pp. 1502–1534
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Solving Strong-Substitutes Product-Mix Auctions

Elizabeth Baldwin,[a] Paul W. Goldberg,[b] Paul Klemperer,[a] Edwin Lock[a,b,*]

[a] Department of Economics, University of Oxford, Oxford OX1 3UQ, United Kingdom; [b] Department of Computer Science, University of Oxford, Oxford OX1 3QD, United Kingdom

*Corresponding author

Contact: elizabeth.baldwin@economics.ox.ac.uk, https://orcid.org/0000-0002-5353-1839 (EB); paul.goldberg@cs.ox.ac.uk, https://orcid.org/0000-0002-5436-7890 (PWG); paul.klemperer@nuffield.ox.ac.uk, https://orcid.org/0000-0003-4372-2962 (PK); edwin.lock@economics.ox.ac.uk, https://orcid.org/0000-0002-0604-2602 (EL)

Received: October 7, 2019
Revised: February 6, 2021; September 8, 2021; November 30, 2022; June 23, 2023
Accepted: July 5, 2023
Published Online in Articles in Advance: August 22, 2023

MSC2020 Subject Classification: Primary: 91-08, 91B26

https://doi.org/10.1287/moor.2019.0248

Copyright: © 2023 INFORMS

Abstract. This paper develops algorithms to solve strong-substitutes product-mix auctions: it finds competitive equilibrium prices and quantities for agents who use this auction's bidding language to truthfully express their strong-substitutes preferences over an arbitrary number of goods, each of which is available in multiple discrete units. Our use of the bidding language and the information it provides contrasts with existing algorithms that rely on access to a valuation or demand oracle. We compute market-clearing prices using algorithms that apply existing submodular minimization methods. Allocating the supply among the bidders at these prices then requires solving a novel constrained matching problem. Our algorithm iteratively simplifies the allocation problem, perturbing bids and prices in a way that resolves tie-breaking choices created by bids that can be accepted on more than one good. We provide practical running time bounds on both price finding and allocation and illustrate experimentally that our allocation mechanism is practical.

Funding: E. Baldwin and P. Klemperer were supported by the Economic and Social Research Council [Grant ES/L003058/1]. P. W. Goldberg and E. Lock were supported by a JP Morgan faculty fellowship during the work on the final version of the paper.

Supplemental Material: The online companion is available at https://doi.org/10.1287/moor.2019.0248.

Keywords: bidding language • competitive equilibrium • Walrasian equilibrium • convex optimization • strong substitutes • submodular minimization • product-mix auction

1. Introduction

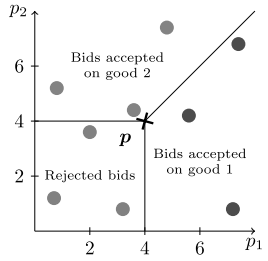
This paper develops algorithms that solve product-mix auctions in which participants can make bids that represent any strong-substitutes (SS) preferences for an arbitrary number of distinct goods. (These preferences are also known, in other literatures, as $M^\natural$-concave, matroidal, and well-layered maps). It, thus, allows bidders to express more general preferences than could previously be permitted in these auctions and finds competitive equilibrium prices and quantities consistent with these and the auctioneer's preferences.

Importantly, our algorithms for finding equilibrium differ from existing ones in that they directly use the information that the product-mix auction "language" provides. This information is in a very different form from the information provided by a valuation or demand oracle. This creates additional complexities as well as simplifications that we can exploit. However, the language is conceptually simple and easy for bidders to use in a (product-mix) auction.

The product-mix auction was developed in 2007–2008 for the Bank of England to provide liquidity to financial institutions by auctioning loans to them (Klemperer [15]). It is now used at least monthly by the Bank and more often when institutions are more likely to be under stress. (After the 2016 vote for Brexit and starting again in March 2019, for instance, the auction was run weekly.)

The original implementation of the auction allowed bidders to each submit a list of bids, on which each bid has $n+1$ elements: a price for each of the n goods available and a total quantity of goods sought by that bid.[1] The auction sets a uniform price for each good (i.e., every recipient of any particular good pays the same price per unit for that good) and gives each bid an allocation that maximizes that bid's "utility," assuming preferences with quasilinear utilities. That is, each bid is allocated its desired quantity of the good on which its price strictly exceeds the auction's price by most, if there is a unique such good, and is allocated nothing if all its prices are strictly below the auction's corresponding prices. (So, if a bid only wants one particular good, it simply sets prices of 0 for all the other $(n-1)$ goods.) A bid that creates a tie (i.e., two or more of its prices exceed the

Figure 1. Price vectors divide $\mathbb{R}^n$ into regions that specify which good is allocated to bids lying in these regions. This example works in the setting with two goods. The price vector $\boldsymbol{p} = (4,4)$ (marked by a cross) divides $\mathbb{R}^2$ into three regions, each of which is labeled according to the good that bids in this region are allocated.

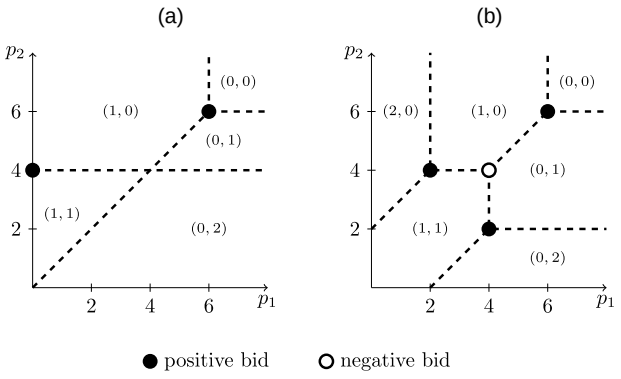


auction's corresponding prices by most, or none of its prices exceed but at least one equals the auction's corresponding prices) can be allocated in any way consistent with equilibrium (see the following). Figure 1 illustrates how the good that a bid is allocated depends on the bid's relative position to the price vector. In particular, we note that individual bids $(b_1,\dots,b_n;+1)$ that seek exactly one item of a good can be interpreted as a unit-demand bidder with valuation vector $\boldsymbol{b} = (b_1,\dots,b_n)$ and quasilinear utilities. We motivate the bidding language in the following example.

Example 1 (Figure 2(a)). Alice participates in an auction held with two goods, say apples (good 1) and bananas (good 2). For her breakfast, she requires a single item of fruit and is willing to pay up to £6 for either an apple or a banana. This is expressed by the bid $\boldsymbol{b} := (6,6;+1)$; the last element in the bid denotes the quantity of goods sought. Suppose the auction's prices are set at $\boldsymbol{p} = (p_1,p_2)$. If at least one good is priced lower than the respective bid entry, she demands the good $i \in \{1,2\}$ that maximizes utility $b_i - p_i$. Otherwise, if both apples and bananas are priced higher than her valuation of £6, she demands neither. In addition, as she is fond of bananas, Alice wishes to pick up a banana as a second fruit item but only if the price of a banana does not exceed £4. This is expressed by the bid $\boldsymbol{b}' := (0,4;+1)$.

To illustrate Alice's demand, consider the following two possible auction prices. If the auction prices are set at $\boldsymbol{p} = (1,3)$, then Alice demands one apple and one banana: $\boldsymbol{b}$ demands an apple as its utility for receiving an apple or a banana is £5 and £3, respectively, whereas $\boldsymbol{b}'$ demands a banana as its price is less than £4. On other hand, if auction prices are $\boldsymbol{p} = (6,5)$, Alice demands only one banana: $\boldsymbol{b}$ demands a banana and $\boldsymbol{b}'$ demands nothing as the price of a banana is too high. Alice's two bids $\boldsymbol{b}$ and $\boldsymbol{b}'$, together with the demand correspondence they induce, are shown in Figure 2(a).

Figure 2. Examples of strong-substitutes demand correspondences on two goods {1, 2} belonging to two bidders, Alice and Bob. Alice's demand is specified by bid list $\mathcal{B} = \{(6,6;+1),(0,4;+1)\}$, and Bob's bid list is $\mathcal{B}' = \{(2,4;+1),(4,2;+1),(4,4;-1),(6,6;+1)\}$. Positive and negative bids are depicted as solid and hollow circles, respectively. Price space is divided into regions corresponding to demanded bundles (x_1, x_2) with x_i denoting the number of items of good i. At $\boldsymbol{p} = (4,4)$, Alice demands bundles $\{(1,0),(0,1),(1,1),(0,2)\}$ and Bob demands bundles $\{(1,0),(0,1),(1,1)\}$, the discrete convex hulls of bundles demanded in the regions surrounding $\boldsymbol{p}$.



Notes. (a) Alice's demand. (b) Bob's demand.

The auction's prices are chosen to maximize the sum of the bids' and auctioneer's welfare; that is, it finds competitive equilibrium prices and quantities.[2] Moreover, the auctioneer expresses preferences about which goods to allocate in the form of supply functions that can themselves be equivalently represented as (and, thus, converted into) a list of bids of the kind described.[3]

1.1. Positive and Negative Bids

In versions of the product-mix auction thus far implemented, all bids are for positive quantities of goods. In the case with transferable utilities, the market-clearing prices can then be found by solving straightforward linear programs, and finding an allocation of the auctioneer's supply to the individual bidders is similarly straightforward. However, some strong-substitutes[4] preferences (over bundles of nonnegative quantities of goods) can only be expressed by using lists of bids for both positive and negative quantities,[5] and it has been shown that any strong-substitutes preferences can be represented using lists of positive and negative bids.[6] The rules for accepting or not accepting negative bids are identical to those for positive bids: if a negative bid is accepted at some prices, it is allocated a negative quantity of the good it demands, (partially) canceling the allocated quantities of positive bids that are accepted at the same prices.

To illustrate why negative bids are useful, we consider the setting with two distinct goods 1 and 2 in which an agent has strong-substitutes valuation u and is interested in at most two units. In other words, the agent only has a positive value for bundles $(0,1),(1,0),(1,1),(0,2)$, and $(2,0)$, where the two-dimensional vectors (x_1, x_2) denote bundles containing x_1 units of good 1 and x_2 units of good 2. It is known (Shioura [37]) that the strong-substitutes condition is equivalent to $M^\natural$-concavity, which, in our setting, is equivalent to discrete concavity together with the following three additional conditions:

1. $u((1,1)) \le u((0,1)) + u((1,0))$,
2. $u((2,0)) - u((1,0)) \le u((1,1)) - u((0,1))$,
3. $u((0,2)) - u((0,1)) \le u((1,1)) - u((1,0))$.

It is easy to see (by infinitesimally and independently perturbing bundle values) that the generic case has strict inequalities in all three conditions. Intuitively, a strict inequality in the first condition implies that a single unit of good 1 is valued strictly less if the agent also has a unit of good 2 than if the agent does not, reflecting strict substitutability as distinct from complementarity or independence between the two goods. A strict inequality in the second or third condition implies that the marginal utility of a good is reduced more by receiving an additional unit of the same good than by receiving a unit of a different good; more variety is valuable. In contrast, it is easy to check that any demand correspondence generated only by positive bids is nongeneric in that at least one of the three conditions must be an equality. Thus, a demand correspondence such as the one illustrated in Example 2 and Figure 2(b), in which all three conditions are strict inequalities, cannot be represented by positive bids only. We refer to Klemperer [15, 17, appendix IC] for further details and examples of the usefulness of negative bids.

Example 2 (Figure 2(b)). A combination of positive and negative bids is required to express Bob's demand correspondence depicted in Figure 2(b), which divides price space into six demand regions for bundles with at most two units. We have $u((0,1)) = u((1,0)) = 6$, $u((0,2)) = u((2,0)) = 8$ and $u((1,1)) = 10$. The corresponding bid list is $\mathcal{B}' = \{(2,4;+1),(4,2;+1),(4,4;-1),(6,6;+1)\}$.

1.2. Our Contributions

This paper addresses the computational challenges of determining uniform component-wise minimal market-clearing prices (at which total market demands equal the quantities of each good that are available) and allocating a fixed collection of the goods at these prices to bidders whose demands are defined by lists of positive and negative bids that express strong-substitutes preferences. Our algorithms exploit the fact that the use of the product-mix bidding language allows for the efficient computation of a demanded bundle as well as of the indirect utility derived at any given prices.

Section 2 introduces the product-mix auction's strong-substitutes bidding language in more detail and develops some of its properties. A first contribution of our paper is to show that it is coNP-complete to determine whether a given list of positive and negative bids constitutes a valid demand correspondence. However, when the number of goods or the number of negative bids is bounded by a constant, we present a polynomial-time algorithm for checking validity.

In Section 3, we consider algorithms for finding component-wise minimal market-clearing prices that have practical running time bounds. We adopt a discrete steepest descent method from the discrete optimization literature (Murota [25, chapter 10], Shioura [37]) that employs submodular minimization to find the steepest descent

directions and discuss two techniques (one novel and one suggested by Shioura [37]) that reduce the number of iterations required by taking long steps in the steepest descent direction. Full details are provided in Appendix B. We show that the resulting algorithms run in time polynomial in the input size of the bid lists, and because submodular minimization is rapid in practice (Chakrabarty et al. [7]), we expect our approach to have a fast running time. Indeed, preliminary experiments (given in the online companion to this paper) suggest our method performs well in practice.

Our main contribution, given in Section 4, is an efficient polynomial-time algorithm that allocates the auctioneer's chosen supply among the bidders at given market-clearing prices. The difficulty in developing this lies in handling bids whose utility is maximized on more than one alternative good (or whose utility from its most preferred good is exactly zero) as tie-breaking choices interact with each other. This gives rise to a novel matching problem. Our algorithm proceeds by iteratively simplifying the allocation problem at hand, allocating unambiguous bids and perturbing bids and prices in a way that resolves a subset of the tie breaks and yields a simplified allocation problem. Progress is measured in terms of reductions in the number of edges of a multigraph associated with the allocation problems.

1.3. Software Implementations

Our price-finding and allocation algorithms are conceptually simple. They are also straightforward to implement, which allowed us to develop two practical implementations in Haskell and Python; these can be found at http://pma.nuff.ox.ac.uk and https://github.com/edwinlock/product-mix, respectively. Some effort was made to optimize for speed in the Python implementation by exploiting fast matrix operations provided by the NumPy package (van der Walt et al. [40]). Furthermore, in an effort to implement submodular minimization efficiently, the Fujishige–Wolfe algorithm (Chakrabarty et al. [6]) was implemented in combination with a memoization technique to reduce the number of submodular function queries.

In order to evaluate the practical running time of our allocation algorithm, we used our Python implementation to run experiments on auctions with various numbers of goods, bidders, and bids. The results of these experiments, given in the online companion to this paper, demonstrate that our allocation algorithm is efficient in practice for realistic auction sizes.

1.4. Related Work

Our work continues a long literature, spanning economics and discrete convex analysis, on gross substitutes (GS) and strong substitutes (Kelso and Crawford [14], Milgrom and Strulovici [24]) and $M^\natural$-concavity (Murota and Shioura [28]). Whereas the discrete convex analysis literature generally allows for multiple units of each good, much focus in economics has been on the case in which there is only one unit of each good; Milgrom and Strulovici [24] show that strong substitutes provide the suitable generalization of gross substitutes to the multiunit case, retaining existence of equilibrium and insisting that any two units of the same good should have the same price.

There is now a substantial body of related work on the two algorithmic problems we address in this paper: finding market-clearing prices and computing an equilibrium allocation of supply to the buyers.

Market-clearing prices are commonly found by either performing an iterative discrete steepest descent search or solving a convex optimization problem with a cutting-plane method.

The method of steepest descent goes back to Ausubel [1], Kelso and Crawford [14], and Murota et al. [31, 32]; see Murota [25, chapter 11; 27] and Paes Leme [33] for recent surveys. However, whereas the steepest descent methods in these references run in pseudo-polynomial time (in the valuation and demand oracle settings), we describe two steepest descent algorithms based on Shioura [37] that take longer steps by exploiting the bid representation of bidder demand and show that they have polynomial running time in the size of the bid lists. (An algorithm is polynomial if its running time is polynomial in the input size and pseudo-polynomial if its running time is required to be polynomial only in numeric values within the input. These differ when numerical values are encoded in binary.)

Baldwin et al. [5] develop a conceptually similar descending-price algorithm using difference of convex functions programming.[7] Extensive computational experiments in Baldwin et al. [5] suggest that this algorithm is competitive with the steepest descent algorithm we describe, and neither algorithm consistently outperforms the other. However, no polynomial running time bound is known for the algorithm of Baldwin et al. [5], whereas we provide a polynomial bound (in the bid list representation size) for our method.

Paes Leme and Wong [34] take the alternative approach of finding market-clearing prices by solving a convex optimization problem. Their solution builds on an improved cutting-plane method of Lee et al. [18] to achieve a polynomial-time algorithm. However, to the best of our knowledge, the cutting-plane method has not yet been

implemented and is computationally expensive in practice; moreover, it is not guaranteed to find componentwise minimal prices.

The harder algorithmic problem is finding a valid allocation of supply to buyers. Most previous studies fail to address this problem in the case when multiple units of each good are available. The two existing algorithms for finding multiunit allocations of which we are aware, in Murota and Tamura [30] and Paes Leme and Wong [34], both work in the valuation oracle model. That is, their algorithms rely on oracle access to the valuation function of each bidder. In order to apply these algorithms in our bidding language setting, we require the computational overhead of simulating a valuation oracle. Section 2.3 describes a method for this simulation, which takes time $O(n^2|\mathcal{B}|^3 + n|\mathcal{B}|T(n))$ per valuation query (or $O(n^2|\mathcal{B}|^2 \log M + n|\mathcal{B}|T(n))$ for an alternative variant) in which n is the number of goods, $|\mathcal{B}|$ is the number of bids, M is an upper bound on the bid entries, and $T(n)$ is the time it takes to minimize an n-dimensional submodular set function.

The allocation algorithm of Murota and Tamura [30], also described in Murota [25, chapter 11], works by reducing the allocation problem to an M-convex network flow problem using access to bidders' valuation functions. Following Murota [25, chapters 10 and 11] and Shioura [36] (for a polynomial-time algorithm for minimizing M-convex functions), this approach requires $O((nm)^6)$ queries to the valuation function, in which m is the number of bidders. With the additional overhead from simulating the valuation oracle, this leads to a running time of $O(m^6n^8|\mathcal{B}|^3 + m^6n^7|\mathcal{B}|T(n))$.

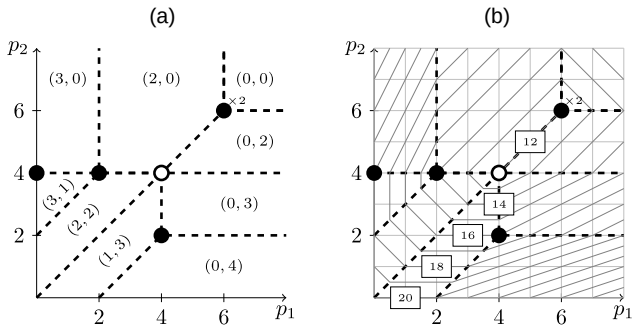
Paes Leme and Wong [34] provide a different algorithm, also in the valuation oracle setting, that solves the allocation problem only if the target bundle is uniquely demanded at some prices. Such prices are guaranteed in the single-unit case (i.e., when there is only one unit of every good) but need not exist in the multiunit case. (Consider, for instance, the aggregate demand of Alice and Bob in Figure 2, also shown in Figure 3(a) of Section 3. The bundle (1, 1) is not uniquely demanded at any prices.) However, if the target bundle is uniquely demanded at some prices, Paes Leme and Wong's [34] algorithm makes $\tilde{O}(mn + n^3)$ queries to the valuation functions to allocate the target bundle. With the simulation of valuation queries, this leads to running-time complexity $\tilde{O}((mn^2 + n^4)(n|\mathcal{B}|^3 + |\mathcal{B}|T(n)))$.

In contrast to the algorithms of Murota and Tamura [30] and Paes Leme and Wong [34], our allocation algorithm directly exploits the specific representation provided by the product-mix auction's bidding language. Theorem 6 establishes that our algorithm takes time $O(mn^3\alpha(n)|\mathcal{B}| + mn^2T(n))$. As the near-constant Ackermann function α satisfies $\alpha(n) \le 4$ for any realistic number of goods n,[8] we see that our algorithm outperforms those of Murota and Tamura [30] and Paes Leme and Wong [34] (when applicable) in the product-mix auction setting. Moreover, our algorithm has been implemented and is efficient in practice.

We note that the valuation oracle setting of Murota et al. [31, 32] and others provides no straightforward way to compute a demanded bundle or the indirect utility function at given prices. In contrast, our bidding-language setting is somewhat analogous to, but more informative than, the demand oracle setting used in Ausubel [1], for example, which presupposes access to an oracle returning a demanded bundle at given prices.

This paper assumes that bidders are able to construct and submit bid lists that represent their strong-substitutes preferences. When the number of goods is large or the preferences are otherwise complex, this may present difficulties for bidders. Goldberg et al. [9] address this issue by presenting algorithms that generate bid lists on behalf of bidders using only access to a demand or valuation oracle to elicit bidders' preferences.

Figure 3. Panel a shows the demand of the aggregate bid list obtained by aggregating Alice's and Bob's bids from Figure 2, whereas panel b depicts a contour plot of the Lyapunov function g for target bundle $\boldsymbol{t} = (1,1)$. Note that $\boldsymbol{p} = (4,4)$ is the minimal market-clearing price for $\boldsymbol{t}$. Starting at $\mathbf{0}$, MinUp repeatedly moves in direction $d = (1,1)$ until it reaches $\boldsymbol{p}$, whereas LongStep-MinUp makes a single long step from $\mathbf{0}$ to $\boldsymbol{p}$.



Notes. (a) Aggregate demand of Alice and Bob. (b) Lyapunov function g for target bundle $\boldsymbol{t} = (1,1)$.

2. Preliminaries

We denote $[n] := \{1, \dots, n\}$ and $[n]_0 := \{0, \dots, n\}$. In our auction model, there are n distinct goods $[n]$; a single copy of a good is an item. A bundle of goods, typically denoted by $\boldsymbol{x}, \boldsymbol{y}$, or $\boldsymbol{z}$ in this paper, is a vector in $\mathbb{Z}^n$ whose ith entry denotes the number of items of good i. The target bundle $\boldsymbol{t}$ is a bundle the auctioneer wants to allocate among the bidders. Vectors $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^n$ typically denote vectors of prices with a price entry for each of the n goods. We write $\boldsymbol{p} \leq \boldsymbol{q}$ when the inequality holds component-wise. It is often convenient to regard a rejected bid as being accepted on a notional reject good for which bids and prices are always zero. Letting the reject good be 0, the set of goods is then $[n]_0$. In this setting, we identify bundles and prices with the $n+1$-dimensional vectors obtained by adding a 0th entry of value 0.

A valuation u is a function that maps (nonnegative) bundles to nonnegative real numbers. The valuations we consider are concave-extensible and have bounded effective domain.[9] Bidders have quasilinear utilities; that is, the utility derived from bundle $\boldsymbol{x}$ at price $\boldsymbol{p}$ by a bidder with valuation u is given by

$$u(\boldsymbol{x}) - \boldsymbol{p} \cdot \boldsymbol{x}. \tag{1}$$

Any valuation u is associated with a demand correspondence D_u that maps prices $\boldsymbol{p}$ to the set of bundles that maximize (1). We omit the subscript u when it is clear from context.

For any subset $X \subseteq [n]$, $\boldsymbol{e}^X$ denotes the characteristic vector of X, that is, an n-dimensional vector whose ith entry is 1 if $i \in X$ and 0 otherwise. Furthermore, $\boldsymbol{e}^i$ denotes the vector whose ith entry is 1 and other entries are 0.

A set function $f : 2^{[n]} \to \mathbb{Z}$ is submodular if it satisfies $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq V$. Submodular function minimization (SFM) is the task of finding a minimizer of a submodular function. It is well-known that the minimizers of submodular functions form a lattice; that is, if S and T are minimizers of f, then so are $S \cup T$ and $S \cap T$. SFM can be solved in polynomial time (Iwata [11], Jiang [12], McCormick [22]). For SFM that is efficient in practice, we refer to two SFM algorithms from the literature. The subgradient descent approach by Chakrabarty et al. [7] determines a minimizer in time $O(nF^3\gamma \log n)$, whereas the Fujishige–Wolfe algorithm described by Chakrabarty et al. [6] takes time $O(F^2(n^2\gamma + n^3))$, where F is an upper bound on the absolute value of f and γ denotes the time it takes to query f. Experimental results (Chakrabarty et al. [6]) indicate that the running time of the Fujishige–Wolfe algorithm depends less on F than suggested by the above bound. For the price-finding algorithms described in Appendix B, we require a subroutine that finds the inclusion-wise minimal minimizer. Note that such a subroutine can be obtained by calling any SFM algorithm $n+1$ times. Indeed, let S be a minimizer of f, and for every $v \in S$, let S_v be a submodular minimizer of the function f restricted to $[n] \setminus \{v\}$. Then, if S_0 denotes the minimal minimizer of f, we have $v \in S_0$ if and only if $f(S_v) > f(S)$ as the minimizers of a submodular function form a lattice. Hence, we obtain $S_0 := \{v \in S \mid f(S_v) > f(S)\}$.

2.1. Strong-Substitutes Valuation Functions

In this paper, bidders have strong-substitutes valuations. We review some basic properties of SS valuations. An SS valuation u divides price space into regions corresponding to bundles: any bundle $\boldsymbol{x}$ has a region of prices in which $\boldsymbol{x}$ is demanded, possibly along with other bundles (see Figure 2). It is known (Murota [25, theorem 11.16]) that each such region is a convex lattice. When a region for some bundle $\boldsymbol{x}$ has full dimensionality, then $\boldsymbol{x}$ is uniquely demanded at any prices in the region's interior; we call this interior a unique demand region (UDR). Moreover, the locus of indifference prices (LIP) $\mathcal{L}_u$ comprising the set of prices at which demand is not unique consists of the union of $(n-1)$-dimensional linear pieces, or facets, that separate the UDRs. In Figure 2, the LIPs associated with the demand correspondences of Alice and Bob in Examples 1 and 2 are illustrated with dashed lines. We refer to Appendix A for further details on the LIP, including how every facet of the LIP can be endowed with a weight to capture the change in demand between any two price vectors.

Definition 1. A valuation u is gross substitutes if, for any prices $\boldsymbol{p}' \geq \boldsymbol{p}$ and any $\boldsymbol{x} \in D_u(\boldsymbol{p})$, there exists $\boldsymbol{x}' \in D_u(\boldsymbol{p}')$ such that $x'_k \geq x_k$ for all k such that $p_k = p'_k$. A valuation u is strong-substitutes (SS) if, when we consider every unit of every good to be a separate good, u is gross substitutes.[10]

Importantly, Shioura and Tamura [38, theorem 4.1] establish that strong substitutes is also equivalent to $M^\natural$-concavity (Murota and Shioura [28]) in our setting of valuations with bounded nonnegative domains. (Both strong substitutes and $M^\natural$-concavity imply concave extensibility; see Milgrom and Strulovici [24, theorem 9] and Murota [25, theorem 6.42], respectively.)

Whereas GS guarantees that a competitive equilibrium exists in single-unit auction markets, the condition is not sufficient for the existence of such an equilibrium in the multiunit case (Shioura and Tamura [38] give an example). SS represents a natural generalization of single-unit GS that provides a general sufficient condition for

the existence of an equilibrium. (See Baldwin and Klemperer [2] and Shioura and Tamura [38] for discussion of the distinction between GS and SS.)

Definition 2. The indirect utility function f_u of valuation u maps a price vector $\boldsymbol{p}$ to the utility that a bidder with demand D_u has for receiving the bidder's preferred bundle at a given price vector $\boldsymbol{p}$ in the following way:

$$f_u(\boldsymbol{p}) := \max_{\boldsymbol{x}}(u(\boldsymbol{x}) - \boldsymbol{p}\cdot\boldsymbol{x}) = u(\boldsymbol{x}') - \boldsymbol{p}\cdot\boldsymbol{x}' \text{ with } \boldsymbol{x}' \in D_u(\boldsymbol{p}). \tag{2}$$

We note that $f_u(\boldsymbol{p})$ is convex, piecewise-linear, and continuous for SS valuations u.

2.2. Representing Strong-Substitutes Valuations with Weighted Bids

We describe how every strong-substitutes valuation function can be represented by a finite list $\mathcal{B}$ of positive and negative bids. A bid $\boldsymbol{b}$ consists of an $(n+1)$-dimensional integral vector $(b_1,\dots,b_n;b_{n+1})$, where the first n components $b_1,\dots,b_n$ of the vector denote the bid's valuation for the goods $[n]$ and the $(n+1)$th component b_{n+1} is interpreted as the bid's weight (the quantity of goods sought by the bid, which may be positive or negative). For the reader's convenience, we also define the alternative notation $w(\boldsymbol{b}) := b_{n+1}$ to denote $\boldsymbol{b}$'s weight. Moreover, we restrict ourselves to positive and negative unit weights ± 1. This is without loss of generality as any bid with a weight of $w(\boldsymbol{b}) \in \mathbb{Z}$ can be represented by $|w(\boldsymbol{b})|$ unit bids with the same vector and of the same sign. Finally, when working with the notional reject good 0 as described in Section 2, we identify the bid $\boldsymbol{b}$ with the $(n+2)$-dimensional vector $(0,b_1,\dots,b_n;b_{n+1})$ obtained by prepending a 0th entry of value 0 (and, thus, indexing from 0 instead of 1). This allows us to define a demand correspondence $D_{\mathcal{B}}$ that maps each price vector $\boldsymbol{p}$ to a set of bundles demanded at $\boldsymbol{p}$ and an indirect utility function $f_{\mathcal{B}}$ associated with $\mathcal{B}$.

A bid $\boldsymbol{b}$ demands good $i \in [n]_0$ at $\boldsymbol{p}$ if the surplus $b_i - p_i$ at price $\boldsymbol{p}$ is maximal, that is, if we have $i \in \arg\max_{i\in[n]_0}(b_i - p_i)$; recall that good 0 is the notional reject good, and $b_0 = p_0 = 0$ by definition. We say that $\arg\max_{i\in[n]_0}(b_i - p_i)$ is the set of demanded goods of $\boldsymbol{b}$ at $\boldsymbol{p}$. A bid $\boldsymbol{b}$ is marginal (on the set of its demanded goods) at $\boldsymbol{p}$ if $\boldsymbol{b}$ demands more than one good at $\boldsymbol{p}$. It is marginal on good i at $\boldsymbol{p}$ if it is marginal at $\boldsymbol{p}$ and i is in its set of demanded goods. Moreover, we say that price $\boldsymbol{p}$ is marginal with respect to (w.r.t.) a given bid list $\mathcal{B}$ if there are bids in $\mathcal{B}$ that are marginal at $\boldsymbol{p}$ and nonmarginal otherwise. We illustrate these definitions using Example 1 (Figure 2(a)). Recall that Alice's bid list $\mathcal{B}$ consists of two positively weighted bids $\boldsymbol{b} := (6,6;1)$ and $\boldsymbol{b}' := (0,4;1)$. At prices $\boldsymbol{p} = (2,4)$, the bid $\boldsymbol{b}$ is not marginal as it demands only good 1, whereas $\boldsymbol{b}'$ is marginal between the reject good and good 2. At prices $\boldsymbol{p} = (6,6)$, the bid $\boldsymbol{b}$ is marginal between the reject good and goods 1 and 2, whereas $\boldsymbol{b}'$ is rejected (it only demands the reject good). Finally, at prices $\boldsymbol{p} = (6,2)$, both bids are nonmarginal as they each demand only good 2. More generally, in Figure 2(a), all prices along the dashed lines are marginal, and all other prices are nonmarginal. Note that these marginal prices coincide with the LIP $\mathcal{L}_u$, that is, the set of prices at which demand is not unique, of Alice's underlying valuation u.

For any bid list $\mathcal{B}$, we define the demand correspondence $D_{\mathcal{B}}(\boldsymbol{p})$ at prices $\boldsymbol{p}$ as follows, distinguishing between the cases in which $\boldsymbol{p}$ is marginal and nonmarginal. If $\boldsymbol{p}$ is nonmarginal, every bid $\boldsymbol{b} \in \mathcal{B}$ uniquely demands some good $i(\boldsymbol{b})$. In this case, $D_{\mathcal{B}}(\boldsymbol{p})$ is a singleton set consisting of the bundle $\boldsymbol{x}$ that is obtained by adding up an amount $w(\boldsymbol{b})$ of good $i(\boldsymbol{b})$ for each $\boldsymbol{b} \in \mathcal{B}$, that is, $D_{\mathcal{B}}(\boldsymbol{p}) = \{\boldsymbol{x}\}$ with $\boldsymbol{x} := \sum_{\boldsymbol{b}\in\mathcal{B}} w(\boldsymbol{b})\boldsymbol{e}^{i(\boldsymbol{b})}$. In Example 1, the demand correspondence at nonmarginal prices $\boldsymbol{p} = (6,2)$ is $D_{\mathcal{B}}(6,2) = \{(0,2)\}$ as both bids demand good 2. At $\boldsymbol{p} = (1,2)$, the demand correspondence is $D_{\mathcal{B}}(1,2) = \{(1,1)\}$ as $\boldsymbol{b}$ uniquely demands good 2 and $\boldsymbol{b}'$ uniquely demands good 1. Figure 2(a) shows the demand in the different polyhedron interiors corresponding to nonmarginal prices.

If $\boldsymbol{p}$ is marginal, $D_{\mathcal{B}}(\boldsymbol{p})$ consists of the discrete convex hull of the bundles demanded at nonmarginal prices arbitrarily close to $\boldsymbol{p}$, at which the discrete convex hull of a set of bundles X is defined as $\mathrm{conv}(X) \cap \mathbb{Z}^n$. In Figure 2(a), we see that bundles demanded in the local neighborhood of $\boldsymbol{p} = (0,4)$ are $(1,0)$ and $(1,1)$, which implies $D_{\mathcal{B}}(\boldsymbol{p}) = \{(1,0),(1,1)\}$. At $\boldsymbol{p} = (4,4)$, we have $D_{\mathcal{B}}(\boldsymbol{p}) = \{(1,0),(0,1),(1,1),(0,2)\}$.

For any bid list $\mathcal{B}$, we can define the indirect utility function

$$f_{\mathcal{B}}(\boldsymbol{p}) = \sum_{\boldsymbol{b}\in\mathcal{B}} w(\boldsymbol{b}) \max_{i\in[n]_0}(b_i - p_i). \tag{3}$$

From (3), it is clear that we can compute $f_{\mathcal{B}}(\boldsymbol{p})$ in linear time in the number of goods and the size of the bid list. In our setting, we can also efficiently compute a bundle demanded at a given price $\boldsymbol{p}$. This demand oracle problem is noted in Paes Leme [33] as an algorithmic primitive needed to implement the Walrasian tâtonnement procedure. If $\boldsymbol{p}$ is nonmarginal, simply allocate each bid $\boldsymbol{b}$ a positive or negative item of the good $i(\boldsymbol{b})$ it uniquely demands and add up these items to obtain the demanded bundle $\boldsymbol{x} = \sum_{\boldsymbol{b}\in\mathcal{B}} w(\boldsymbol{b})\boldsymbol{e}^{i(\boldsymbol{b})}$. However, if $\boldsymbol{p}$ is marginal and the bid list contains both positive and negative bids, we emphasize that independently allocating to each bid one of the goods it demands may not result in a bundle that lies in $D_{\mathcal{B}}(\boldsymbol{p})$. In Example 2 (depicted in Figure 2(b)),

for instance, we see that, at prices $\boldsymbol{p} = (4,4)$, allocating a negative item of good 1 to bid $(4,4;-1)$, a positive item of good 1 to bid $(4,2;1)$, and a positive item of good 2 each to bids $(2,4;1)$ and $(6,6;1)$ leads to a total bundle of $(0,2) \notin D_{\mathcal{B}}(\boldsymbol{p})$. Instead, care must be taken to accept bids in a consistent way; one way to do this is to perturb entries of the price vector slightly so as to break ties and then note that the resulting bundle is demanded at the unperturbed prices.

Concurrent work by Baldwin and Klemperer [3] (see also Endnote 6) shows that any SS demand correspondence can be represented as a finite list of positive and negative bids, and this representation is essentially unique (up to redundancies). Conversely, however, not all lists of positive and negative bids induce a valuation function. We call a bid list valid if the indirect utility function $f_{\mathcal{B}}$ defined in (3) is convex; Theorem 1 gives two further equivalent characterizations of validity.[11] In particular, it establishes that the indirect utility function and demand correspondence of a valid bid list as defined in Section 2.2 coincide with the indirect utility function and demand correspondence for some strong-substitutes valuation as defined in the preceding sections. In Section 4, we also introduce a weaker local notion of validity that holds in an open, convex neighborhood of a price $\boldsymbol{p}$.

Theorem 1. *Let $\mathcal{B}$ be a list of positive and negative bids and $f_{\mathcal{B}}$ be its associated indirect utility function as defined in (3). The following conditions are equivalent:*

1. *$f_{\mathcal{B}}$ is convex.*
2. *For every price vector $\boldsymbol{p}$ and pair of distinct goods $i, i' \in [n]_0$, it holds that $\sum_{\boldsymbol{b} \in \mathcal{B}_{ii'}} w(\boldsymbol{b}) \geq 0$, where $\mathcal{B}_{ii'} \subseteq \mathcal{B}$ is the subset of bids marginal on both i and i' at $\boldsymbol{p}$.*
3. *There exists a strong-substitutes valuation u satisfying $f_{\mathcal{B}} = f_u$ and $D_{\mathcal{B}} = D_u$.*

The equivalence of (1) and (2) is a special case of Proposition 1 in Section 4, which is proved in Appendix A.2. It is well-known that (3) implies (1). To show that (2) implies (3), we use results of Mikhalkin [23] to verify the existence of a suitable valuation u, and check that u indeed has the properties required. We present the full proof of Theorem 1 in Appendix A.1.

2.3. Simulating a Valuation Oracle

A bid list $\mathcal{B}$ does not directly provide the value $u(\boldsymbol{x})$ that the bidder has for a given bundle $\boldsymbol{x}$. However, the ability to evaluate $u(\boldsymbol{x})$ for arbitrary bundles is a prerequisite for the allocation algorithms in Murota [26] and Paes Leme and Wong [34] as mentioned in the related work section. An efficient way to determine $u(\boldsymbol{x})$ is to compute prices $\boldsymbol{p}$ at which the bidder demands $\boldsymbol{x}$ using the price-finding algorithm from Section 3; then, computing the bidder's indirect utility $f_u(\boldsymbol{p}) = f_{\mathcal{B}}(\boldsymbol{p})$ (cf. Definitions 2 and 3) at these prices; and finally, recovering the bundle value $u(\boldsymbol{x}) = f_{\mathcal{B}}(\boldsymbol{p}) + \boldsymbol{p} \cdot \boldsymbol{x}$. The first step, finding the prices $\boldsymbol{p}$, takes time $O(n^2|\mathcal{B}|^3 + n|\mathcal{B}|T(n))$ with the demand change method (and $O(n^2|\mathcal{B}|^2 \log M + n|\mathcal{B}|T(n))$ with the binary search method) for determining the step lengths of the steepest descent method described in Section 3 and Appendix B. Here, n is the number of goods, M is an upper bound on the bid entries, and $T(n)$ is the time it takes to minimize an n-dimensional submodular set function. The second and third steps, computing $f_{\mathcal{B}}(\boldsymbol{p})$ and solving for $u(\boldsymbol{x})$, take time $O(n|\mathcal{B}|)$, so the first step dominates the computation time.

2.4. Deciding Validity of Bid Lists

In our auction, bidders submit their lists of positive and negative bids to the auctioneer prior to the auction. We show in Theorem 2 that checking the validity of a given list of bids is computationally hard. The proof, given in Appendix C, exploits definition (2) of validity in Theorem 1. However, in Appendix C, we provide a straightforward algorithm to verify the validity of a given bid list in polynomial time if the number of goods or the number of negative bids is bounded by a constant.

Theorem 2. *Deciding the validity of a given list of positive and negative bids is coNP-complete.*

Theorem 3. *Let $\mathcal{B}$ be a list of positive and negative bids. If the number of goods or the number of negative bids is bounded by a constant, there exists a polynomial-time algorithm for checking the validity of $\mathcal{B}$.*

Furthermore, many useful subclasses of bid lists (of arbitrary size) are easily checkable for validity in practice. Hence, whereas the coNP-completeness result is disappointing, it does not seem to be an important limitation on the auction as sensible restrictions on the permitted bids can allow us to check validity efficiently in practice.

2.5. The Computational Challenges

In this section, we state the computational problems to be solved. For any bidder $j \in J$, where J is the set of bidders, $\mathcal{B}^j$ denotes the bids of bidder j. We assume each $\mathcal{B}^j$ is valid as defined in Section 2.2 and provided as a list

of vectors encoded in binary. Let $\mathcal{B}$ be the aggregate list of bids obtained by concatenating the lists of all bidders. Theorem 1, part (2), implies that $\mathcal{B}$ is valid. Figure 3(a) depicts the aggregate bid list of Alice and Bob from Figure 2. The running times of our algorithms are given in terms of n, $|J|$, $|\mathcal{B}|$, and $M := \max_{\boldsymbol{b}\in\mathcal{B}}\|\boldsymbol{b}\|_\infty$, the maximum bid vector entry.

Suppose the auctioneer intends to sell target bundle $\boldsymbol{t}$. Our aim is to compute a competitive equilibrium: a market-clearing price $\boldsymbol{p}$ at which $\boldsymbol{t}$ is demanded and an allocation of $\boldsymbol{t}$ to the various bidders so that every bidder receives a bundle demanded at $\boldsymbol{p}$. In the event that not enough bids are made for the target bundle, some items can go unsold, which is equivalent to the auctioneer buying them back from the market. To reflect this, the auctioneer places a total of $\|\boldsymbol{t}\|_1 + 1$ positive bids at $\mathbf{0}$.[12] The computation of a competitive equilibrium separates into two problems.

The Price-Finding Problem. Given the aggregate bid list $\mathcal{B}$ and target bundle $\boldsymbol{t}$, find the coordinate-wise minimal market-clearing price $\boldsymbol{p}$ at which $\boldsymbol{t}$ is demanded, that is, $\boldsymbol{t} \in D_{\mathcal{B}}(\boldsymbol{p})$.

The Allocation Problem. Given a valid bid list $\mathcal{B}^j$ for each bidder $j \in J$, a target bundle $\boldsymbol{t}$, and market-clearing price $\boldsymbol{p}$, allocate $\boldsymbol{t}$ to the bidders so that each bidder receives a bundle demanded at $\boldsymbol{p}$, that is, a partition $(\boldsymbol{t}^j)_{j\in J}$ of $\boldsymbol{t}$ with $\boldsymbol{t}^j \in D_{\mathcal{B}^j}(\boldsymbol{p})$ for all $j \in J$.

Note that we do not ask for a breakdown of which of j's bids are accepted on which goods. Indeed, we show that a solution to the allocation problem can be obtained without this knowledge.

Theorem 4. *Suppose that each bidder $j \in J$ has SS demand correspondence D^j. Given a target allocation $\boldsymbol{t}$ and a market-clearing price $\boldsymbol{p}$ for $\boldsymbol{t}$, there exists a partition $\boldsymbol{t} = \sum_{j\in J}\boldsymbol{t}^j$ such that $\boldsymbol{t}^j \in D^j(\boldsymbol{p})$.*

Theorem 4 (cf. Danilov et al. [8], Milgrom and Strulovici [24], Murota [25]) ensures that, if the target bundle $\boldsymbol{t}$ is aggregately demanded at market-clearing price $\boldsymbol{p}$, then there exists an allocation of $\boldsymbol{t}$ among the bidders so that every bidder receives a bundle the bidder demands. However, care must be taken in finding such an allocation as bidders cannot simply be allocated an arbitrary bundle they demand at the market-clearing price. (Indeed, in Figure 2, the lowest prices at which Alice and Bob aggregately demand (1, 1) is given by $\boldsymbol{p} = (4, 4)$. If we allocate Alice the whole bundle (1, 1), Bob does not demand the empty remainder (0, 0) at $\boldsymbol{p}$.)

If we had (oracle) access to the valuations of the bidders, we could perturb the valuations such that the target bundle is uniquely demanded at some price by subtracting carefully constructed functions that are discrete-convex in one variable.[13] However, it is not clear how to perturb the bids individually to achieve a suitable indirect perturbation of the valuations. Instead, our allocation algorithm takes the approach of perturbing bids bidder by bidder. Some intuition can be gained by considering the addition of a small random perturbation vector $\boldsymbol{v}^j$ to all the bids in bid set $\mathcal{B}^j$ (a different $\boldsymbol{v}^j$ for each $j \in J$) and recomputing the new market-clearing prices. This has the effect of breaking ties between bids in two different bid lists. (This also uses the plausible fact, established in Proposition 2, that small perturbations make only small changes to the market clearing price(s).) However, the perturbations need to be exponentially small in the number of goods in order to ensure that all possible ties are broken. (One can check, noting the pigeonhole principle, that if perturbations are multiples of some inverse-polynomial, then they will have distinct subsets having the same sum, losing the guarantee that all possible ties are broken.) Our algorithm of Section 4 avoids this by systematically perturbing and unperturbing bid sets, making decisions on marginal bids as it proceeds.

3. Finding Market-Clearing Prices

In order to determine component-wise minimal prices at which the market is cleared, we apply an iterative steepest descent approach from the discrete convex optimization literature (Shioura [37]). This approach generalizes an algorithm used by Ausubel's [1] ascending auction design and Gul and Stacchetti [10] for the task of finding equilibrium prices in single-unit markets. Ausubel [1] and others define the Lyapunov function with regard to indirect utility function f_u for strong-substitutes valuation u and target bundle $\boldsymbol{t}$ as $g_{\boldsymbol{t}}(\boldsymbol{p}) := f_u(\boldsymbol{p}) + \boldsymbol{t}\cdot\boldsymbol{p}$.

A function $f : \mathbb{Z}^n \to \mathbb{R}$ is $L^\natural$-convex if it satisfies the translation submodularity property,

$$f(\boldsymbol{p}) + f(\boldsymbol{q}) \geq f((\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q}) + f(\boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1})) \quad (\forall \boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}^n_+,\ \forall \alpha \in \mathbb{Z}_+). \tag{4}$$

Here, $\vee$ and $\wedge$ denote the component-wise maximum and minimum, respectively. It is well-known that the Lyapunov function $g_{\boldsymbol{t}}$ is $L^\natural$-convex if the valuation is strong substitutes (see Murota et al [31, 32] or Proposition B.1 in Appendix B) and, thus, has a unique component-wise minimal minimizer (Shioura [37]). Moreover, this minimizer is integral and corresponds to the component-wise minimal market clearing prices (cf. Lemma B.1 in Appendix B) and can be found using the following discrete steepest descent procedure. Given a starting point

$\boldsymbol{p} := \boldsymbol{p}^0$ that is dominated by the market-clearing prices $\boldsymbol{p}^*$, repeatedly find the component-wise minimal subset $S \subseteq [n]$ minimizing $g_t(\boldsymbol{p} + \boldsymbol{e}^S) - g_t(\boldsymbol{p})$ and move a unit step in this direction by updating $\boldsymbol{p} := \boldsymbol{p} + \boldsymbol{e}^S$. The method terminates once a local minimum is reached. As a local minimum of an $L^\natural$-convex function is a global minimum (see, e.g., Shioura [37],), we have found $\boldsymbol{p}^*$. It is straightforward that the $L^\natural$-convexity of the Lyapunov function as defined in (4) guarantees the submodularity of function $g'(\boldsymbol{p}; S) := g_t(\boldsymbol{p} + \boldsymbol{e}^S) - g_t(\boldsymbol{p})$ with respect to argument S (for any fixed $\boldsymbol{p}$). Hence, we can apply standard submodular minimization algorithms (cf. Section 2) to find the component-wise minimal steepest descent direction at any prices $\boldsymbol{p}$.

Adapting this method to our bidding-language setting, we can use our direct knowledge of the bids and (3) to express g_t for any valid bid list $\mathcal{B}$ as

$$g_t(\boldsymbol{p}) := f_{\mathcal{B}}(\boldsymbol{p}) + \boldsymbol{t} \cdot \boldsymbol{p} = \sum_{\boldsymbol{b} \in \mathcal{B}} w(\boldsymbol{b}) \max_{i \in [n]_0} (b_i - p_i) + \boldsymbol{t} \cdot \boldsymbol{p}. \tag{5}$$

Theorem 1 guarantees that the function $f_{\mathcal{B}}$ defined in (3) is the indirect utility function f_u of some strong-substitutes valuation u, so (5) is a well-defined Lyapunov function. (Figure 3(b) depicts a contour plot of the Lyapunov function defined for the aggregate demand of Alice and Bob as given in Examples 1 and 2 and shown in Figure 3(a).) Moreover, (5) implies that g_t can be evaluated at any $\boldsymbol{p}$ in time $O(n|\mathcal{B}|)$.

In order to improve the running time of the steepest descent method, Shioura [37] proposes aggregating several steps in the same direction into a single "long" step. After finding the component-wise minimal direction $\boldsymbol{e}^S$, we move $\lambda(\boldsymbol{p}, S)$ unit steps in this direction at once by updating $\boldsymbol{p} := \boldsymbol{p} + \lambda(\boldsymbol{p}, S)\boldsymbol{e}^S$, where $\lambda(\boldsymbol{p}, S)$ is defined as

$$\lambda(\boldsymbol{p}, S) := \max\{\lambda \in \mathbb{Z}_+ \mid g'(\boldsymbol{p}; S) = g'(\boldsymbol{p} + (\lambda - 1)\boldsymbol{e}^S; S)\} \tag{6}$$

and $g'(\boldsymbol{p}; S) := g_t(\boldsymbol{p} + \boldsymbol{e}^S) - g_t(\boldsymbol{p})$. Shioura [37] shows that the number of long steps before reaching $\boldsymbol{p}^*$ is at most $n\max\{-g'(\boldsymbol{0}; S) \mid S \subseteq [n]\}$. In our setting, we see that $-g'(\boldsymbol{0}; S) \leq |\mathcal{B}|$ and obtain an upper bound of $n|\mathcal{B}|$ on the number of long steps that our steepest descent method takes (see Appendix B).

In Appendix B, we describe two methods for computing the step length (6) at every iteration of the steepest descent procedure and analyze their complexity in our bidding-language setting. The first method uses binary search, as suggested by Shioura [37], to find each $\lambda(\boldsymbol{p}, S)$; we show that determining each $\lambda(\boldsymbol{p}, S)$ takes time $O(n|\mathcal{B}|\log M)$. (Recall that $M = \max_{\boldsymbol{b} \in \mathcal{B}} \|\boldsymbol{b}\|_\infty$.) Our second method for determining $\lambda(\boldsymbol{p}, S)$ (the demand change method) directly exploits our knowledge of the individual bids to determine the step length in time $O(n|\mathcal{B}|^2)$. We see that the best method in practice depends on the magnitudes of $\log M$ and $|\mathcal{B}|$. Overall, both methods lead to price-finding algorithms that are polynomial in the size of the bid lists as we state in Theorem 5.

Theorem 5. *The steepest descent approach takes time $O(n^2|\mathcal{B}|^2 \log M + n|\mathcal{B}|T(n))$ and $O(n^2|\mathcal{B}|^3 + n|\mathcal{B}|T(n))$ to find the component-wise smallest market-clearing prices using the binary search and demand change long step methods, respectively.*

The computation time of the steepest descent method, even with long steps, is dominated by the task of finding a component-wise minimal submodular minimizer. In practice, we can exploit our knowledge of the bids to decrease the dimensionality of the submodular function to be minimized, which leads to performance improvements. For full details on these practical improvements and on the unit-step and long-step steepest descent methods for computing market-clearing prices in the bidding-language setting, we refer to Appendix B.

4. Allocations to the Separate Bidders

Suppose we are given a market-clearing price $\boldsymbol{p}$ for target bundle $\boldsymbol{t}$. We now present an algorithm that solves the allocation problem, that is, finds a partition $(\boldsymbol{t}^j)_{j \in J}$ of the target bundle $\boldsymbol{t}$ such that $\boldsymbol{t}^j$ is demanded by bidder j at price $\boldsymbol{p}$. We note that, whereas the market-clearing price $\boldsymbol{p}$ returned by our steepest descent approach in Section 3 is component-wise minimal, our allocation algorithm works for any integral market-clearing price.

Algorithm 1 (ALLOCATE)

Input: Initial allocation problem $\mathcal{A}$.
Output: Target bundle allocation.
Allocate all nonmarginal bids by applying NONMARGINALS (Algorithm 3).
while $\mathcal{A}$ has a nonempty bid list **do**
  Run the FINDPARAMS subroutine (Algorithm 2).
  if FINDPARAMS returns a demand cluster (I, j^*) with at most one link good i^* **then**
    Process unambiguous marginal bids with UNAMBIGUOUSMARGINALS (Algorithm 4).
  else

9: Simplify the allocation problem with ShiftProjectUnshift (Algorithm 5).
10: Allocate all (newly) nonmarginal bids with NonMarginals (Algorithm 3).
11: **return** the partial allocation bundle $\boldsymbol{m}^j$ for each bidder $j \in J$.

Our algorithm Allocate, stated in Algorithm 1, repeatedly simplifies the problem until it becomes vacuous. It generates a sequence of allocation problems by iteratively allocating parts of the target bundle to bidders and removing satisfied bids from the bid lists until the residual supply bundle (initially the target bundle) is empty. In order to capture this formally, we define a generalized allocation problem in Section 4.1; the definition introduces a weaker notion of local validity for bid lists and also features a partial allocation bundle for each bidder as well as a residual target bundle. With every allocation problem, we associate a corresponding marginal bids multigraph and derived graph. These graphs and key terminology, such as demand clusters and link goods, are defined in Section 4.2. The marginal bids multigraph is used to quantify progress and establish the running time of the algorithm, whereas the derived graph is required for Allocate to decide whether to apply the UnambiguousMarginals or ShiftProjectUnshift procedure.

The algorithm starts by removing all nonmarginal bids and transferring their demanded items from the residual supply to the partial allocation bundle of the bid's owner. This is performed by the procedure NonMarginals, which is defined in Section 4.3. Next, the algorithm calls the FindParams subroutine (defined in Section 4.2), which constructs and works on the derived graph. If FindParams identifies a subset of marginal bids that can be allocated unambiguously, Allocate invokes UnambiguousMarginals (defined in Section 4.3) to process these bids. Otherwise FindParams identifies parameters that Allocate uses to call the ShiftProjectUnshift procedure. ShiftProjectUnshift is defined in Section 4.4; it does not allocate items and delete bids, but instead simplifies the allocation problem by shifting and projecting the bids in order to strictly reduce the demand ties between bids. Finally, NonMarginals is invoked again to process any bids that may have become nonmarginal. The algorithm repeats this process until the allocation problem become vacuous in the sense that all bid lists are empty, at which point the partial allocation bundles constitute a valid allocation of the target bundle $\boldsymbol{t}$ at price vector $\boldsymbol{p}$. Section 4.5 gives the running time and a proof of correctness for Allocate.

4.1. The Generalized Allocation Problem

We now introduce allocation problems formally and define the terminology and technical tools required to reason about how Allocate successively reduces the initial allocation problem using the three procedures NonMarginals, UnambiguousMarginals, and ShiftProjectUnshift. In general, we note that applying these procedures may result in bid lists that are no longer valid in the sense of Theorem 1. (Figure 5 in Section 4.3 gives an example of how applying NonMarginals can result in an invalid bid list.) Instead, we introduce the notion of local validity that requires bid lists to be valid only in the neighborhood of the market-clearing price $\boldsymbol{p}$. For prices $\boldsymbol{p}$ in this neighborhood, we can define the demand correspondence $D_{\mathcal{B}}(\boldsymbol{p})$ in the same way as for globally valid bid lists. Let $B(\boldsymbol{p}, \varepsilon) := \{\boldsymbol{y} \in \mathbb{R}^n \mid \|\boldsymbol{y} - \boldsymbol{p}\|_\infty < \varepsilon\}$ denote the open ε-ball with regard to the L_∞-norm.

Definition 3. A list of bids $\mathcal{B}$ is ε-valid at $\boldsymbol{p}$ if the indirect utility function $f_{\mathcal{B}}$ restricted to $B(\boldsymbol{p}; \varepsilon)$ is convex. The bid list is locally valid at $\boldsymbol{p}$ if there exists some $\varepsilon > 0$ for which it is ε-valid.

This generalizes the notion of valid bid lists introduced in Section 2.2, and we correspondingly generalize Theorem 1 with the following proposition (proved in Appendix A.2).

Proposition 1. *Let $\mathcal{B}$ be an ε-valid bid list at $\boldsymbol{p}$ and let $P = B(\boldsymbol{p}; \varepsilon)$. The following are equivalent:*

1. *The indirect utility function $f_{\mathcal{B}}$ restricted to P is convex.*
2. *There is no price vector $\boldsymbol{p} \in P$ and pair of distinct goods $i, i' \in [n]_0$ at which the weights $w(\boldsymbol{b})$ of the bids $\boldsymbol{b} \in \mathcal{B}$ marginal on i and i' sum to a negative number.*

Definition 4. An allocation problem is a four-tuple $\mathcal{A} = [\boldsymbol{p}, (\mathcal{B}^j)_{j \in J}, (\boldsymbol{m}^j)_{j \in J}, \boldsymbol{r}]$, where

1. $\boldsymbol{p} \in \mathbb{R}^{n+1}$ is a price vector with $p_0 = 0$.
2. For each bidder $j \in J$, $\mathcal{B}^j$ is a list of bids in $\{0\} \times (\mathbb{Z} + \{0, 1/10\})^n \times \mathbb{Z}$ that is locally valid at $\boldsymbol{p}$.
3. $\boldsymbol{m}^j \in \mathbb{Z}_+^{n+1}$ for each $j \in J$ is the partial allocation.
4. $\boldsymbol{r} \in \mathbb{Z}_+^{n+1}$ is the residual supply,

and there exists a valid allocation $(\boldsymbol{r}^j)_{j \in J}$ of the residual supply $\boldsymbol{r}$ to bidders, that is, $\boldsymbol{r}^j \in D_{\mathcal{B}^j}(\boldsymbol{p})$ and $\sum_{j \in J} \boldsymbol{r}^j = \boldsymbol{r}$. For any such valid allocation, we say that $\boldsymbol{t}^j := \boldsymbol{r}^j + \boldsymbol{m}^j$ is a solution to $\mathcal{A}$.

Note that the bids need not be integral, and the bid lists for the bidders are only required to be locally valid at $\boldsymbol{p}$. The intuition behind this definition is to capture the action of successively allocating items of the target bundle $\boldsymbol{t}$ to bidders, which may break global validity but preserves local validity at $\boldsymbol{p}$. For each bidder $j \in J$, the partial allocation $\boldsymbol{m}^j$ is a bundle denoting the subset of $\boldsymbol{t}$ already allocated to j, whereas $\boldsymbol{r}$ denotes the remaining part of $\boldsymbol{t}$

not yet allocated to any bidder. The 0th coordinates of $\boldsymbol{m}^j$ and $\boldsymbol{r}$ denote the number of reject goods (not yet) allocated.

Definition 5. We say that $\mathcal{A}'$ is a valid reduction from $\mathcal{A}$ if $\mathcal{A}'$ is an allocation problem and all solutions to $\mathcal{A}'$ are also solutions to $\mathcal{A}$.

In the initial problem, the residual supply is given by the target bundle, and the partial allocation vectors are $\mathbf{0}$. Note that t_0 denotes the total number of rejected bids and can be computed as the total weight of bids minus the total number of items in $\boldsymbol{t}$. In the vacuous problem, the bid lists are empty, and the residual supply is $\mathbf{0}$.

4.1.1. Technical Tools. The following observations and lemma are used in Section 4.4 to prove that applying the procedure ShiftProjectUnshift to an allocation problem $\mathcal{A}$ produces an allocation problem that is a valid reduction from $\mathcal{A}$. We define the surplus gap of a bid $\boldsymbol{b}$ at $\boldsymbol{p}$ as the difference between the utility derived from a demanded good and the maximum utility derived from a nondemanded good. This allows us to describe the goods that $\boldsymbol{b}$ demands when we perturb the price or the bid by a small amount. Moreover, we see in Lemma 1 that the size of the neighborhood around $\boldsymbol{p}$ in which a bid list is valid can be lower bounded by the surplus gaps of the bids in $\mathcal{B}$. Formally, let $I = \arg\max_{i\in[n]_0} b_i - p_i$ denote the goods demanded by $\boldsymbol{b}$ at $\boldsymbol{p}$. Then, if $I \neq [n]_0$, we define the surplus gap of $\boldsymbol{b}$ at $\boldsymbol{p}$ as

$$\max_{i\in[n]_0}(b_i - p_i) - \max_{i\in[n]_0\setminus I}(b_i - p_i).$$

Observation 1. If bid $\boldsymbol{b}$ demands goods I at $\boldsymbol{p}$ with a surplus gap of at least ε, then for any price $\boldsymbol{p}' \in B(\boldsymbol{p}, \varepsilon/2)$, the goods I' demanded by $\boldsymbol{b}$ at $\boldsymbol{p}'$ form a subset of I.

Observation 2. If bid $\boldsymbol{b}$ demands goods I at $\boldsymbol{p}$ with a surplus gap of at least ε, then for any bid $\boldsymbol{b}' \in B(\boldsymbol{b}, \varepsilon/4)$ and price $\boldsymbol{p}' \in B(\boldsymbol{p}, \varepsilon/4)$, the goods I' demanded by $\boldsymbol{b}'$ at $\boldsymbol{p}'$ form a subset of I.

Lemma 1. *If $\mathcal{B}$ is δ-valid at $\boldsymbol{p}$ for some $\delta > 0$ and all bids in $\mathcal{B}$ have a surplus gap of at least ε at $\boldsymbol{p}$, then $\mathcal{B}$ is $\varepsilon/2$-valid at $\boldsymbol{p}$.*

We prove this lemma in Appendix A.3.

4.2. The Marginal Bids Graph and the Derived Graph

With every allocation problem $\mathcal{A}$, we associate a marginal bids multigraph and a derived graph. As mentioned, the marginal bids multigraph allows us to provide upper bounds on the running time of our allocation algorithm. Moreover, in order to decide whether to apply UnambiguousMarginals or ShiftProjectUnshift and to determine the input parameters for these procedures, our Allocate algorithm calls the subroutine FindParams, defined in this section, which first constructs the derived graph and then attempts to find a maximal open or closed path in this graph.

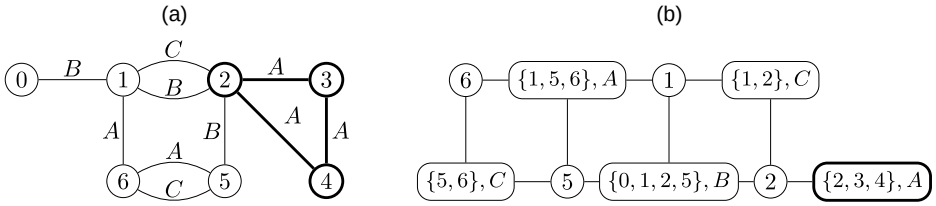
Definition 6. The marginal bids graph $G_{\mathcal{A}}$ associated with allocation problem $\mathcal{A}$ is an undirected edge-labeled multigraph whose vertices are the goods $[n]_0$ (including the reject good). The graph $G_{\mathcal{A}}$ has an edge (i, i') labeled with j if bidder j has a bid that is marginal at $\boldsymbol{p}$ between i and i'.

For any bidder j, let $G_{\mathcal{A}}[j]$ denote the subgraph of $G_{\mathcal{A}}$ induced by all j-labeled edges; note that this graph is simple as there is at most one edge labeled j connecting two goods i and i' in $G_{\mathcal{A}}$. A pair (I, j) with $I \subseteq [n]_0$ and $j \in J$ is a demand cluster if I is the set of vertices of some connected component of $G_{\mathcal{A}}[j]$. Intuitively, this captures the dependencies between the demands of (and, thus, the possible allocations to) bidder j's bids. A vertex is a link good if its incident edges are labeled with more than one bidder. (Effectively, a link good expresses a dependency between demand clusters of different bidders.) We call a cycle in $G_{\mathcal{A}}$ a multibidder cycle if it contains edges labeled by different bidders. A vertex in such a cycle is called a cycle-link good if its two edges in the cycle are labeled differently. Note that any multibidder cycle has at least two cycle-link goods. As with link goods, multibidder cycles and their cycle-link goods capture the dependencies between demands of (and, thus, possible allocations to) different bidders.

Definition 7. The derived graph $D_{\mathcal{A}}$ associated with allocation problem $\mathcal{A}$ is a simple bipartite graph whose two disjoint and independent vertex sets are the set of link goods and the set of demand clusters of $G_{\mathcal{A}}$. There is an edge between link good i and demand cluster (I, j) if $i \in I$.

Note that, by the definition of link goods, every link good is adjacent to at least two demand clusters in the derived graph. This is consistent with our observation that a link good can be interpreted as a link between connected components of $G_{\mathcal{A}}[j]$ for different bidders j. Figure 4 gives an example of a marginal bids multigraph and its corresponding derived graph.

Figure 4. Example of a marginal bids multigraph (panel a) and derived graph (panel b) with three bidders A, B, C and six goods $1, \ldots, 6$ together with reject good 0. Goods $1, 2, 5, 6$ are link goods. Goods 2, 3, 4 form a demand cluster $(\{2, 3, 4\}, A)$ with one link good represented by a leaf demand cluster in the derived graph (bolded in both graphs).



Notes. (a) A marginal bids multigraph. (b) A derived graph.

We describe a procedure to construct the derived graph in near-linear time $O(\alpha(n)n|\mathcal{B}|)$ using a disjoint-union data structure. Here, $\alpha(\cdot)$ is the inverse Ackermann function, which grows extremely slowly and is near-constant in our context as $\alpha(n) \le 4$ for any $n \le 2^{2^{2^{16}}}$. The disjoint-union data structure (cf. Tarjan and van Leeuwen [39]) maintains a representation of a set partition and admits an $\alpha(n)$-time operation to merge two subsets of the partition. Internally, it assigns a distinct label to each subset of the partition and provides $\alpha(n)$-time access to the label of the subset in which a given element lies.

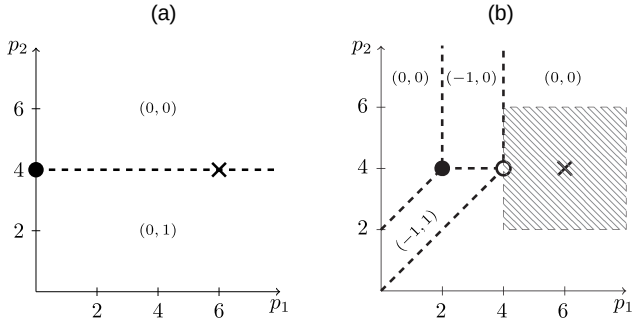
First, we show how to compute the demand clusters for each bidder. Fix a bidder j and initialize the disjoint-union data structure. For each bid $\boldsymbol{b} \in \mathcal{B}^j$, compute the marginal goods S at prices $\boldsymbol{p}$ and, fixing any $i \in S$, merge the sets containing i and i' for all $i' \in S \setminus \{i\}$. Now, the data structure has learned the vertex partition induced by the connected components of $G_{\mathcal{A}}$ in time $O(\alpha(n)n|\mathcal{B}^j|)$. In order to recover this partition and express it as a family of sets, we initialize an empty family $\mathcal{K}$ of sets, in which each set in $\mathcal{K}$ has an associated label. For each good $i \in [n]$, determine the label l of i's subset in the data structure. If there already is a set in $\mathcal{K}$ with label l, add i to this set. Otherwise, add the new singleton set $\{i\}$ with label l to $\mathcal{K}$. Finally, iterate through $\mathcal{K}$ and delete all singleton sets. This takes time $O(\alpha(n)n)$. Now, $\mathcal{K}$ represents the family of demand clusters of bidder j. Hence, in total, it takes time $O(\alpha(n)n|\mathcal{B}|)$ to compute the demand clusters for all bidders.

In order to compute the link goods, iterate through the demand clusters of all bidders and count the number of times each good appears. If a good appears at least twice, it is a link good. Once we know which good is a link good, we can compute the edges of the derived graph: for each demand cluster (I, j), add an edge between (I, j) and each link good in I. Iterating through the demand clusters of all bidders takes $O(nm) = O(n|\mathcal{B}|)$ time.

4.2.1. The FindParams Subroutine.

In every iteration, the Allocate algorithm employs a subroutine to decide, using the derived graph, whether to invoke UnambiguousMarginals or ShiftProjectUnshift and to compute the input parameters for the respective procedure. This subroutine FindParams, given in Algorithm 2, takes as input an allocation problem and returns one of three possible outputs: a demand cluster (I, j) with no link goods, a demand cluster (I, j) with one link good denoted i^*, or a cycle-link good i^* and the label j^* of one of its incident

Figure 5. The resulting bid lists after applying NonMarginals (Algorithm 3) to Alice's and Bob's bid lists from Figure 2 at prices $\boldsymbol{p} = (6, 4)$ (marked by a cross). Note that Alice's bid list remains globally valid, whereas Bob's bid list is only locally 2-valid at $\boldsymbol{p}$ (indicated by the hatched square).



Notes. (a) Alice's bid list after applying NonMarginals. (b) Bob's list after applying NonMarginals.

edges in the multibidder cycle. The ALLOCATE algorithm invokes UNAMBIGUOUSMARGINALS if FINDPARAMS returns a demand cluster and SHIFTPROJECTUNSHIFT if it returns a cycle-link good and edge label.

FINDPARAMS works by walking through the derived graph $D_{\mathcal{A}}$ to find a maximal path. A path in a graph is a sequence of distinct vertices $v_1, \dots, v_k$ such that v_i and v_{i+1} are connected by an edge for all $i \in [k-1]$; it is maximal in a given graph if we cannot extend the path with any vertex v_{k+1} of the graph. A path is closed if there is an edge between v_1 and v_k and open otherwise. (Hence, a closed path is a cycle.) Note that a path in $D_{\mathcal{A}}$ alternates between link good and demand cluster vertices.

Algorithm 2 (FINDPARAMS)
1: Compute the derived graph $D_{\mathcal{A}}$.
2: **if** $D_{\mathcal{A}}$ contains an isolated demand cluster **then**
3: **return** this demand cluster (without a link good).
4: Starting from any link good, take a walk in $D_{\mathcal{A}}$ to find a maximal path.
5: Let i^* and (I, j^*) be the last link good and demand cluster visited, respectively.
6: **if** the path is open **then**
7: **return** (I, j^*) and i^*.
8: **else**
9: **return** i^* and j^*.

Lemma 2. *In time $O(\alpha(n)n|\mathcal{B}|)$, FINDPARAMS returns a demand cluster (I, j^*) with no link goods, a demand cluster (I, j^*) with one link good i^*, or a cycle-link good i^* and the label j^* of one of its incident edges in the multibidder cycle.*

Proof. Note that any path in the derived graph $D_{\mathcal{A}}$ has length at most $2n$. Hence, constructing the derived graph, which takes time $O(\alpha(n)n|\mathcal{B}|)$, dominates the running time. By construction of the derived graph, an isolated demand cluster in $D_{\mathcal{A}}$ contains no link goods, whereas a leaf demand cluster contains exactly one link good. Hence, if there is an isolated demand cluster, the subroutine returns a demand cluster without a link good. If the path found is open, the last vertex must be a leaf demand cluster, and the subroutine returns a demand cluster with single link good. Now, suppose the path is closed and consider only the cycle formed by alternating link good and demand cluster vertices. First, note that there is a path between consecutive link goods in the marginal bids graph $G_{\mathcal{A}}$ using only j-labeled edges. Second, consecutive demand clusters have different bidder labels. These two observations imply that $G_{\mathcal{A}}$ contains a simple multibidder cycle with cycle-link good i^* and an incident edge labeled with j^*. □

4.3. Allocating Unambiguous Bids

4.3.1. Nonmarginal Bids. Suppose bidder j has a nonmarginal bid $\boldsymbol{b}$ at market-clearing prices $\boldsymbol{p}$ that demands good $i \in [n]_0$ in the allocation problem $\mathcal{A}$. Then, this bid contributes exactly $w(\boldsymbol{b}) \in \{-1, 1\}$ items of good i to any solution of $\mathcal{A}$. Hence, we can unambiguously allocate these items to $\boldsymbol{m}^j$ and remove them from the residual supply $\boldsymbol{r}$, thus accepting the nonmarginal bid on the appropriate good. NONMARGINALS, given in Algorithm 3, processes all nonmarginal bids in this way. Note that, whereas this operation may not preserve global validity of bid lists, the resulting lists remain locally valid at $\boldsymbol{p}$ so that the result is a valid allocation problem. Figure 5 gives an example. ALLOCATE calls NONMARGINALS in every iteration in order to process nonmarginal bids that can be allocated unambiguously.

Algorithm 3 (NONMARGINALS (Accept Nonmarginal Bids))
1: **Input:** Allocation problem $\mathcal{A} = [\boldsymbol{p}, (\mathcal{B}^j)_{j\in J}, (\boldsymbol{m}^j)_{j\in J}, \boldsymbol{r}]$.
2: **Output:** Reduced allocation problem $\mathcal{A}'$ without nonmarginal bids.
3: **for all** bidders $j \in J$ **do**
4: **for all** nonmarginal bids $\boldsymbol{b} \in \mathcal{B}^j$ **do**
5: Determine unique good i demanded by $\boldsymbol{b}$ and remove $\boldsymbol{b}$ from $\mathcal{B}^j$.
6: Increment m_i^j by $w(\boldsymbol{b})$.
7: Decrement r_i by $w(\boldsymbol{b})$.

Lemma 3. *Given a valid allocation problem $\mathcal{A}$, NONMARGINALS (Algorithm 3) outputs a reduction $\mathcal{A}'$ of $\mathcal{A}$ in linear time. Moreover, we have $G_{\mathcal{A}} = G_{\mathcal{A}'}$.*

Proof. Let $\mathcal{A}' = [\boldsymbol{p}, (\mathcal{B}^j)'_{j\in J}, (\boldsymbol{m}^j)'_{j\in J}, \boldsymbol{r}']$ be the output of NONMARGINALS. First, we show that $\mathcal{A}'$ is a valid allocation problem. Fix some $j \in J$. Because $\mathcal{B}^j$ is locally valid by assumption, the indirect utility function $f_{\mathcal{B}^j}$ is convex in some small neighborhood of $\boldsymbol{p}$. Allocating a nonmarginal bid to bidder j corresponds to subtracting from the

utility function $f_{\mathcal{B}^j}(q)$ the term $\max_{i\in[n]_0}(b_i - q_i)$. In a sufficiently small neighborhood of $\boldsymbol{p}$, this term is linear, so the resulting utility function is also convex in some open neighborhood of $\boldsymbol{p}$.

It is straightforward to see that $(\boldsymbol{t}^j)_{j\in J}$ is a solution to $\mathcal{A}'$ if and only if it is a solution to $\mathcal{A}$, so $\mathcal{A}'$ is a reduction of $\mathcal{A}$. To see that $G_{\mathcal{A}} = G_{\mathcal{A}'}$, note that the marginal bids are unchanged. □

4.3.2. Unambiguous Marginal Bids. Let $\mathcal{A}$ be an allocation problem without nonmarginal bids and suppose FindParams returns a demand cluster (I, j) with at most one link good. UnambiguousMarginals is invoked by Allocate to unambiguously allocate the bids specified by this demand cluster. By the definition of a demand cluster, none of bidder j's bids is marginal between items in both I and $[n]_0 \setminus I$. Let $\mathcal{B}^j_I$ denote the bids marginal on goods in I. All bids $\boldsymbol{b} \in \mathcal{B}^j_I$ contribute an item of a good in I to bidder j's final allocation bundle, whereas all other bids contribute an item of a good not in I.

If the demand cluster (I, j) has no link goods, none of the bids by bidders other than j is marginal on I, so all items of goods $i \in I$ in the residual supply must be allocated to j. Hence, we reduce our allocation problem by transferring r_i items from the residual supply $\boldsymbol{r}$ to bidder j's partial allocation bundle $\boldsymbol{m}^j$ for each $i \in I$ and removing all bids marginal on items in I from $\mathcal{B}^j$.

If the demand cluster (I, j) has a single link good i^*, all items of goods in $I \setminus \{i^*\}$ must be allocated to j by the same argument as earlier. Second, the bids in $\mathcal{B}^j_I$ must be allocated a total of $\sum_{\boldsymbol{b}\in\mathcal{B}^j_I} w(\boldsymbol{b})$ units of goods from I. Hence, the difference $\sum_{\boldsymbol{b}\in\mathcal{B}^j_I} w(\boldsymbol{b}) - \sum_{i\in I\setminus\{i^*\}} r_i$ gives us the number of items of i^* that must be allocated to bidder j. This yields Algorithm 4.

Algorithm 4 (UnambiguousMarginals (Process All Unambiguous Marginal Bids))

Input: Allocation problem $\mathcal{A} = [\boldsymbol{p}, (\mathcal{B}^j)_{j\in J}, (\boldsymbol{m}^j)_{j\in J}, \boldsymbol{r}]$ and demand cluster (I, j) with at most one link good i^*.
Output: Valid reduction $\mathcal{A}'$ of $\mathcal{A}$ with no bids marginal on goods in I.
if (I, j) has no link goods **then**
  for all $i \in I$ **do**
    Increment m^j_i by r_i and set r_i to 0.
else
  Compute $d = \sum_{\boldsymbol{b}\in\mathcal{B}^j_I} w(\boldsymbol{b}) - \sum_{i\in I\setminus\{i^*\}} r_i$.
  Increment $m^j_{i^*}$ by d and decrement r_{i^*} by d.
  for all $i \in I \setminus \{i^*\}$ **do**
    increment m^j_i by r_i and set r_i to 0.
Remove all bids marginal on I from $\mathcal{B}^j$.

Lemma 4. *UnambiguousMarginals, given in Algorithm 4, returns a valid reduction of the input allocation problem in time $O(n|\mathcal{B}^j|)$. Moreover, the marginal bids graph $G_{\mathcal{A}'}$ of $\mathcal{A}'$ has strictly fewer edges than $G_{\mathcal{A}}$.*

Proof. In order to see that $\mathcal{A}'$ is a valid allocation problem, we verify that the new bid list $(\mathcal{B}^j)'$ of bidder j after applying UnambiguousMarginals is locally valid by checking criterion (2) of Definition 3. Fix a price $\boldsymbol{p}$ and goods i, i'. Recall that bids in $\mathcal{B}^j$ cannot be marginal on goods in I and $[n]_0 \setminus I$ by the definition of demand clusters. If $i \in I$ and $i' \in [n]_0 \setminus I$, then $\mathcal{B}^j$ has no bids marginal on i and i'. If both i, i' are goods in I, all bids marginal on i and i' are removed by UnambiguousMarginals. If neither i nor i' is a good in I, then none of the bids marginal on i and i' is removed. As $\mathcal{B}^j$ is locally valid by assumption, this implies that $(\mathcal{B}^j)'$ is also locally valid. It is straightforward to see that a solution to $\mathcal{A}'$ is a solution to $\mathcal{A}$ as the partial allocation performed by UnambiguousMarginals is unambiguous. Finally, let $i, i' \in I$ and note that the marginal bids graph $G_{\mathcal{A}}$ contains an edge between i and i' that is not present in $G_{\mathcal{A}'}$. As removing bids does not introduce new edges to the graph, $G_{\mathcal{A}'}$ has strictly fewer edges. □

4.4. The Shift-Project-Unshift Reduction

Suppose that FindParams returns a cycle-link good i^* and the label j^* of one of its adjacent edges in the cycle. In this case, Allocate invokes the ShiftProjectUnshift procedure to obtain a reduction. Unlike NonMarginals and UnambiguousMarginals, this procedure does not make progress by allocating items and deleting the satisfied bids. Instead, it temporarily shifts the bids of bidder j^* and perturbs the market-clearing price $\boldsymbol{p}$ using SFM. We show that the marginal bids graph for this new allocation problem has strictly fewer edges, reducing the overall dependencies between the demands of bids. Finally, the procedure projects the bids of bidder j^* in such a way that it can unperturb the market-clearing price, retaining the dependency structure of the new marginal bids graph. We now introduce the shift and project operations.

4.4.1. Shifting Bids. Suppose all bids are integral and we shift some bidder's bids by a small quantity $\varepsilon < 1/4$ (we use $\varepsilon = 1/10$ for concreteness) in the direction of i by adding $\varepsilon \boldsymbol{e}^i$ to each bid vector. Then, Lemma 5 and Proposition 2 (proved in Appendix A.4) together show that we can use SFM to find a price $\boldsymbol{p}^\varepsilon \in \{\boldsymbol{p} + \varepsilon \boldsymbol{e}^S, S \subseteq [n]\}$ at which the new allocation problem $\mathcal{A}'$ with shifted bids and price vector $\boldsymbol{p}^\varepsilon$ is a valid reduction.

Lemma 5. *Fix ε with $|\varepsilon| < 1/4$ as well as $i \in [n]$ and $j \in J$. Let $\mathcal{A}$ be an allocation problem with integral bids and prices, and let $\mathcal{A}'$ be obtained by replacing $\mathcal{B}^j$ with $(\mathcal{B}^j)' := \{\boldsymbol{b} + \varepsilon \boldsymbol{e}^i \mid \boldsymbol{b} \in \mathcal{B}^j\}$. Then, the bid lists of all bidders in $\mathcal{A}'$ are locally valid at any price $\boldsymbol{p}^\varepsilon \in B(\boldsymbol{p}, \varepsilon)$. Moreover, the bundles demanded by any bidder at $\boldsymbol{p}^\varepsilon$ in $\mathcal{A}'$ form a subset of the bundles they demand at $\boldsymbol{p}$ in $\mathcal{A}$.*

Proof. As the bids and prices in $\mathcal{A}$ are integral, each bid either demands all goods $[n]_0$ or has a surplus gap of at least 1 at $\boldsymbol{p}$. By Lemma 1, all bid lists are $1/2$-valid at $\boldsymbol{p}$. This implies that every unshifted bid list is $(1/2 - \varepsilon)$-valid at $\boldsymbol{p}^\varepsilon$ and the shifted bid list is $(1/2 - 2\varepsilon)$-valid at $\boldsymbol{p}^\varepsilon$. By Observations 1 and 2, a nonmarginal bid demands the same good for all $\boldsymbol{q} \in B(\boldsymbol{p}^\varepsilon, \delta)$ and sufficiently small $\delta > 0$. Hence, a bidder demands the same set of bundles at all nonmarginal prices in $B(\boldsymbol{p}^\varepsilon, \delta)$. As the bundles a bidder demands at $\boldsymbol{p}^\varepsilon$ are, by definition, the discrete convex hull of bundles they demand at nonmarginal prices infinitesimally close to $\boldsymbol{p}^\varepsilon$, we are done. □

Proposition 2. *Let $\mathcal{A}$ be an allocation problem with price vector $\boldsymbol{p}$ and fix ε with $|\varepsilon| < 1/4$. Suppose we shift all bids of bidder j's in $\mathcal{B}$ by ε to obtain (aggregate) bid list $\mathcal{B}'$. Then, there exists a price $\boldsymbol{p}^\varepsilon \in \{\boldsymbol{p} \pm \varepsilon \boldsymbol{e}^S, S \subseteq [n]\}$ at which the residual supply $\boldsymbol{r}$ is demanded by $\mathcal{B}'$. We can determine $\boldsymbol{p}^\varepsilon$ using submodular minimization on the Lyapunov function g with regard to list $\mathcal{B}'$ and the residual supply $\boldsymbol{r}$.*

4.4.2. Projecting Bids. Next, we define a projection operation on bids. The idea of this operation is to modify a bid $\boldsymbol{b}$ so that its surplus gap at $\boldsymbol{p}$ is increased. As a consequence, if that bid or the price is perturbed slightly, it "remembers" its preferred goods. Let $\boldsymbol{b}$ be a bid and $I = \arg\max_{i \in [n]_0}(b_i - p_i)$ be the set of demanded goods at price $\boldsymbol{p}$. The projection $\boldsymbol{b}'$ of $\boldsymbol{b}$ w.r.t. $\boldsymbol{p}$ is defined as follows:

$$\boldsymbol{b}' := \begin{cases} \boldsymbol{b} - \boldsymbol{e}^{[n]_0 \setminus I} & \text{if } 0 \in I, \\ \boldsymbol{b} + \boldsymbol{e}^I & \text{otherwise.} \end{cases}$$

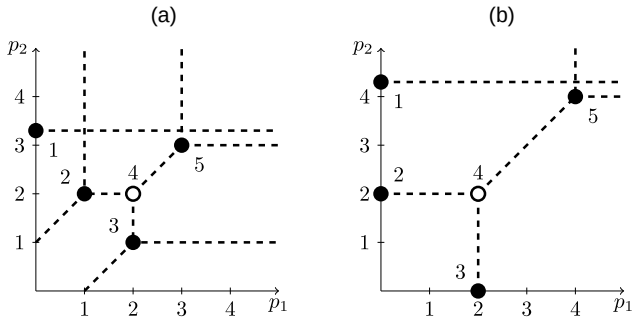
Note that we allow for bid vector entries to be negative. Figure 6 illustrates the projection operation.

Observation 3. If bid $\boldsymbol{b}$ demands all goods $[n]_0$, we have $\boldsymbol{b}' = \boldsymbol{b}$. Otherwise, the projection operation on $\boldsymbol{b}$ w.r.t. $\boldsymbol{p}$ increases the bid's surplus gap at $\boldsymbol{p}$ by 1.

Lemma 6. *Projecting all bids in a bidder's bid list $\mathcal{B}$ w.r.t. $\boldsymbol{p}$ does not change the set of bundles demanded at $\boldsymbol{p}$. Moreover, if $\mathcal{B}$ is locally valid at $\boldsymbol{p}$, the projected bid list is locally $1/2$-valid.*

Proof. Suppose all bids in $\mathcal{B}$ have a surplus gap of at least ε at $\boldsymbol{p}$ and $\mathcal{B}$ is locally δ-valid at $\boldsymbol{p}$. Note that, at any price $\boldsymbol{q}$ in the open ball $B(\boldsymbol{p}, \varepsilon)$ centered at $\boldsymbol{p}$, every bid demands the same set of goods before and after projecting. Hence, the projected bid list is locally $\min\{\delta, \varepsilon\}$-valid at $\boldsymbol{p}$, and the set of bundles demanded at $\boldsymbol{p}$, consisting of the discrete convex hull of bundles demanded in UDRs bordering $\boldsymbol{p}$ remains the same. The second statement follows from Observations 1 and 3. □

Figure 6. Example of the project operation w.r.t. prices $\boldsymbol{p} = (4, 4)$ on five bids numbered from 1 to 5 shown in panel a. The projected bids are shown in panel b. Note that only the negative bid is unchanged as it demands all goods, including the reject good.



Notes. (a) Five numbered bids before projecting. (b) The result of projecting w.r.t. prices $\boldsymbol{p} = (4, 4)$.

4.4.3. The Procedure. The procedure ShiftProjectUnshift is stated in Algorithm 5. It shifts and projects to reduce the number of edges in a marginal goods graph. This effectively reduces the number of demand ties among the bids in our allocation problem. Lemmas 7 (proved in Appendix A.5) and 8 establish that the reduction obtained by ShiftProjectUnshift is valid and makes progress.

Algorithm 5 (ShiftProjectUnshift)
1: **Input:** Allocation problem $\mathcal{A} = [\boldsymbol{p}, (\mathcal{B}^j)_{j\in J}, (\boldsymbol{m}^j)_{j\in J}, \boldsymbol{r}]$, cycle-link good i^*, and edge label j^*.
2: **Output:** Allocation problem $\mathcal{A}'$.
3: Add $\frac{1}{10}\boldsymbol{e}^{i^*}$ to each of j^*'s bids and compute a new market-clearing price $\boldsymbol{p}^\varepsilon = \boldsymbol{p} + \frac{1}{10}\boldsymbol{e}^{S^*}$ by solving $S^* = \arg\min_{S\subseteq[n]} g_{\boldsymbol{r}}\left(\boldsymbol{p} \pm \frac{1}{10}\boldsymbol{e}^S\right)$ using SFM.
4: Project every bid list w.r.t. $\boldsymbol{p}^\varepsilon$.
5: Subtract $\frac{1}{10}\boldsymbol{e}^{i^*}$ from each of j^*'s bids and reset price to $\boldsymbol{p}$.

Lemma 7. *The marginal bids graph $G_{\mathcal{A}'}$ of $\mathcal{A}'$ has strictly fewer edges than $G_{\mathcal{A}}$.*

Lemma 8. *ShiftProjectUnshift, given in Algorithm 5, returns a reduction $\mathcal{A}'$ of $\mathcal{A}$ in time $O(T(n) + n|\mathcal{B}|)$, where $T(n)$ is the time it takes to minimize an n-dimensional submodular function.*

Proof. Let $\mathcal{A}'$ and $\mathcal{A}''$ denote the allocation problems after executing lines 3 and 4, respectively. First, we show that $\mathcal{A}'$ is a reduction of $\mathcal{A}$. Indeed, by Proposition 2 and step 1, the residual bundle $\boldsymbol{r}$ is aggregately demanded at $\boldsymbol{p}^\varepsilon$ in $\mathcal{A}'$. Furthermore, for any allocation $(\boldsymbol{r}^j)_{j\in J} = (\boldsymbol{t}^j - \boldsymbol{m}^j)_{j\in J}$ of the residual bundle $\boldsymbol{r}$ at price $\boldsymbol{p}^\varepsilon$ in $\mathcal{A}'$, Lemma 5 implies that $(\boldsymbol{m}^j + \boldsymbol{r}^j)_{j\in J}$ is a solution of $\mathcal{A}$. Second, Lemma 6 implies that $\mathcal{A}''$ is a reduction of $\mathcal{A}'$. Finally, after line 5, the bid lists are locally valid at $\boldsymbol{p}$ by Lemma 5. Proposition 2 implies that the residual bundle $\boldsymbol{r}$ is demanded at some price $\boldsymbol{p}' \in \{\boldsymbol{p}^\varepsilon \pm \varepsilon \boldsymbol{e}^S \mid S \subseteq [n]\}$. Note that, if a bundle is demanded at $\boldsymbol{p}'$, it is also demanded at $[\boldsymbol{p}']$ because of the structure of our integral bid vectors, where $[\cdot]$ denotes the operation of rounding each component to the nearest integer. As $[\boldsymbol{p}'] = \boldsymbol{p}$, the result follows. □

4.5. The Main Algorithm

The algorithm Allocate, stated in Algorithm 1, combines the procedures NonMarginals, UnambiguousMarginals, and ShiftProjectUnshift in order to solve the allocation problem described in Section 2.5. Recall that it uses FindParams as a subroutine to decide whether to call UnambiguousMarginals or ShiftProjectUnshift in each iteration of the loop. Theorem 6 proves correctness and gives a running time bound for Allocate. We note that this bound is likely to be pessimistic.

Theorem 6. *Allocate solves the allocation problem in time $O(n^2|J|(\alpha(n)n|\mathcal{B}| + T(n)))$, where $T(n)$ is the time required to minimize an n-dimensional submodular set function.*

Proof. By construction, the marginal bids graph of the initial allocation problem has at most $|J|\binom{n+1}{2}$ edges. Every call to UnambiguousMarginals or ShiftProjectUnshift strictly reduces the number of edges (by Lemmas 4 and 7). Hence, after at most $|J|\binom{n+1}{2}$ iterations of step 2, the marginal bids graph of the current allocation problem is an empty graph, implying that there are no more marginal bids. In particular, at this point, a single call to NonMarginals allocates all remaining nonmarginal bids and returns a vacuous allocation problem. As all three procedures NonMarginals, UnambiguousMarginals, and ShiftProjectUnshift return a reduction in the sense of Definition 5, the solution to the final vacuous allocation problem is also a solution to the original allocation problem. To see the running time guarantee, note that FindParams and the procedures NonMarginals, UnambiguousMarginals, and ShiftProjectUnshift are each called at most $|J|\binom{n+1}{2} = O(n^2|J|)$ times, and FindParams dominates NonMarginals and UnambiguousMarginals. □

4.5.1. Incorporating Priorities. The FindParams subroutine as stated in Section 4.2 is not fully specified, and different implementations may lead to different inputs for UnambiguousMarginals or ShiftProjectUnshift when given the same allocation problem. If the input for ShiftProjectUnshift depends on the implementation used, ties may be broken differently, and thus, the target bundle is allocated differently among the bidders.

In order to control which input for ShiftProjectUnshift is returned, we propose the use of a priority list consisting of a permutation of all good-bidder pairs $(i, j) \in [n]_0 \times J$. The priority list can be chosen to favor certain types of bidder; for example, if it is considered desirable to bring "small" bidders (having low demand) into the market, we can prioritize their bids, making it slightly more likely that they will be allocated. Moreover, the list may

be given as an additional input parameter at the start of Allocate or be generated and updated dynamically as Allocate runs. The subroutine FindParams is replaced by PriorityParams, which returns the highest pair (i, j) in the priority list that constitutes a valid input to ShiftProjectUnshift or an input to UnambiguousMarginals if no such pair exists. Recall that a pair (i, j) is a valid input for ShiftProjectUnshift if i is a cycle-link good in some multibidder cycle and j is the label of one of its edges in this cycle. This is the case if the derived graph $D_{\mathcal{A}}$ has a cycle containing the edge from i to the demand cluster (I, j) satisfying $i \in I$. For any (undirected) graph G and edge vw, one can check whether G has a cycle containing vw by determining whether its endpoints v and w are connected in the graph $G - vw$ obtained by deleting vw, using breadth first search (BFS). This implies the following subroutine.

Algorithm 6 (PriorityParams)

1: Compute the derived graph $D_{\mathcal{A}}$.
2: **for all** pairs (i, j) in the priority list **do**
3: **if** i is a link good and $i \in I$ for some demand cluster (I, j) **then**
4: Check whether the edge from i to (I, j) lies in a cycle by temporarily removing the edge and verifying (using BFS) whether the two endpoints are still connected in the graph.
5: **return** (i, j) if the edge lies in a cycle.
6: **return** some leaf demand cluster (I, j) and the adjacent link good i.

To see that PriorityParams is well-defined, note that the priority list is a permutation of all possible pairs (i, j). Hence, if the derived graph contains a cycle, the subroutine returns it; otherwise, there exist at least two leaves for step 2 to choose from. In practice, we can prune the priority list dynamically by removing pairs (i, j) once i is no longer a link good. This is possible because UnambiguousMarginals and ShiftProjectUnshift do not add edges to the derived graph, and so a good cannot become a link good again at some later point.

5. Conclusions and Further Work

This paper provides a practical process for running auctions in which bidders can express strong-substitutes preferences using positive and negative bids. Recent working papers (Baldwin and Klemperer [3], Lin and Tran [19]) show that all strong-substitutes preferences can be represented using appropriate combinations of these bids. Although we show that verifying the validity of bid lists is coNP-complete, useful subclasses of collections of bids are efficiently checkable for validity in practice and can be checked off-line, bidder by bidder, prior to the auction.

Our approach to finding market-clearing prices builds on the long-step steepest descent procedure proposed by Shioura [37]. We both analyze his binary search method for determining long steps in our product-mix auction bidding language context and also propose a novel second method for determining step lengths.

The structure of the product-mix bidding language also facilitates the design of an efficient algorithm for computing an equilibrium allocation of goods to bidders. Our approach contrasts with the previous literature on the problem of allocating supply to bidders at given prices (Murota [25], Murota and Tamura [30], Paes Leme and Wong [34]), which instead assumes that bidders' preferences are accessed through an abstract valuation oracle.

All our algorithm approaches are polynomial time in the size of the bid list inputs and, as we show in the online companion, efficient in practice.

One question for future research is whether we can also exploit the information provided by the bidding language to improve the computational efficiency of the preexisting submodular function minimization subroutine that both our price-finding and allocation algorithms use. Another obvious question is the extent to which our methods can be extended to broader classes of valuations.

Acknowledgments

The authors thank the reviewers for valuable comments and suggestions, which have been helpful in improving this paper. The authors are particularly grateful to a referee who both identified a problem in an earlier draft of the paper and took the trouble to develop an elegant solution to it. The authors also thank colleagues—including Meg Meyer, Duncan Coutts, Martin Bichler, and Maximilian Fichtl—and Tobias Dammers and Andres Löh, who undertook the Haskell implementation and provided much helpful feedback as the algorithms were developed.

Appendix A. Additional Proofs

We give some definitions that are used in the following proofs. Here and in the following, we use terminology and material developed in Baldwin and Klemperer [2], drawing on the literature in tropical geometry, especially Mikhalkin [23]; see also the textbooks by Maclagan and Sturmfels [20] and Joswig [13].

Geometric objects for valuations: The locus of indifference prices (LIP) $\mathcal{L}_u$ for valuation u with demand correspondence D_u is defined as

$$\mathcal{L}_u = \{\boldsymbol{p} \in \mathbb{R}^n \,|\, |D_u(\boldsymbol{p})| > 1\}.$$

This set is identified as an object of mathematical interest by Mikhalkin [23] and is also known as a "tropical hypersurface" (Joswig [13], Maclagan and Sturmfels [20]).

Recall that a rational polyhedral complex is a finite collection of polyhedrons, closed under taking faces; nonempty intersections of these polyhedrons must be faces of them, and these faces must have rational slope (Joswig [13], Maclagan and Sturmfels [20]). $\mathcal{L}_u$ can be decomposed into an $(n-1)$-dimensional rational polyhedral complex Π_u (Mikhalkin [23, proposition 2.1]). We call the $(n-1)$-dimensional polyhedra in this complex its facets. Let w_u be the weight function obtained as follows: if $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the two bundles demanded on either side of a facet F, then the weight $w_u(F)$ of F is the greatest common divisor of the entries of $\boldsymbol{x} - \boldsymbol{y}$. We recall the following proposition from Baldwin and Klemperer [2].

Proposition A.1 (Baldwin and Klemperer [2, proposition 2.4]). *The change in demand as prices change between the UDRs on either side of facet F is $w_u(F)$ times the primitive integer vector that is normal to F and points in the opposite direction to the change in price.*

In particular, we identify the strong-substitutes vectors: those nonzero vectors in $\mathbb{Z}^n$ that have at most one +1 entry, at most one −1 entry, and no other nonzero entries. (These vectors are also distinguished in the literature on discrete convex analysis, being the edges of discrete generalized polymatroids; see, e.g., Shioura and Tamura [38, theorem 3.9]). Recall that we assume throughout this paper that valuations are concave-extensible. The next theorem follows.

Theorem A.1 (Shioura and Tamura [38, theorems 4.1 and 4.10]; See Also Baldwin and Klemperer [2, proposition 3.8]). *Valuation u is strong substitutes if and only if the facets of $\mathcal{L}_u$ are normal to strong-substitutes vectors.*

Following Mikhalkin [23, definition 3], we say that a weighted polyhedral complex (Π, w) is balanced if, for every $(n-2)$-cell G of Π, the weights $w(F_j)$ on the facets $F_1, \ldots, F_l$ that contain G and primitive integer normal vectors $\mathbf{v}_{F_j}$ for these facets that are defined by a fixed rotational direction about G satisfy $\sum_{j=1}^{l} w(F_j)\mathbf{v}_{F_j} = 0$. Balancing is essential to what Baldwin and Klemperer [2, theorem 2.14] call the "valuation-complex equivalence theorem."

Theorem A.2 (Mikhalkin [23, remark 2.3 and proposition 2.4]). *Suppose that (Π, w) is an $(n-1)$-dimensional weighted rational polyhedral complex in $\mathbb{R}^n$ with positive weights only and $\mathcal{L}$ is the union of the cells in Π. There exists a finite set $A \subsetneq \mathbb{Z}^n$ and a concave-extensible valuation $u : A \to \mathbb{R}$ such that $\mathcal{L}_u = \mathcal{L}$ and $w_u = w$ if and only if (Π, w) is balanced. Moreover, in this case, then, for any price vector $\boldsymbol{p}$ not contained in $\mathcal{L}$, there exists a unique concave-extensible such u satisfying also $u(\mathbf{0}) = 0$ and $D_u(\boldsymbol{p}) = \{\mathbf{0}\}$.*

Geometric objects for bids: In analogy to the preceding definitions, we now define geometric objects for product-mix bids and bid lists. The LIP for a bid $\boldsymbol{b}$ is

$$\mathcal{L}_{\boldsymbol{b}} = \left\{ \boldsymbol{p} \in \mathbb{R}^n \,|\, |\arg\max_{i \in [n]_0}(b_i - p_i)| > 1 \right\}.$$

Similarly, the LIP of a list of bids $\mathcal{B}$ is given by

$$\mathcal{L}_{\mathcal{B}} = \{\boldsymbol{p} \in \mathbb{R}^n \,|\, |D_{\mathcal{B}}(\boldsymbol{p})| > 1\}.$$

Any $\mathcal{L}_{\mathcal{B}}$ can be decomposed into an $(n-1)$-dimensional rational polyhedral complex $\Pi_{\mathcal{B}}$. We call the $(n-1)$-dimensional polyhedra in this complex its facets. Endow every facet F of $\Pi_{\mathcal{B}}$ with a weight as follows:

$$w_{\mathcal{B}}(F) := \sum \{w(\boldsymbol{b}) \,|\, \boldsymbol{b} \in \mathcal{B} \text{ and } \mathcal{L}_{\boldsymbol{b}} \supseteq F\}.$$

A.1. Proof of Theorem 1

Proof of Theorem 1. The equivalence of (1) and (2) follows from Proposition 1 if we set $P = \mathbb{R}^n$. The implication (3) ⇒ (1) is well-known; Shioura and Tamura [38, theorem 7.1] and Murota et al. [32, theorem 4.3], for instance, show that f_u is a polyhedral $L^\natural$-convex function, and these functions are defined as a subclass of convex functions (cf. Murota et al. [32]). We now show (2) ⇒ (3) using Theorem A.2. Let $(\Pi_{\mathcal{B}}, w_{\mathcal{B}})$ be the weighted polyhedral complex associated with $\mathcal{B}$. Note that, as well as showing that balancing holds, we must also demonstrate that all weights of the polyhedral complex are positive, which is the setting for Theorem A.2.

Observe first that every facet of $\Pi_{\mathcal{B}}$ has a strong-substitutes vector as its normal vector as this property holds for any individual $\mathcal{L}_{\boldsymbol{b}}$. Moreover, the weighted polyhedral complex $(\Pi_{\boldsymbol{b}}, w_{\boldsymbol{b}})$ associated with a single bid is balanced. If $w(\boldsymbol{b}) = 1$, then this follows because $(\Pi_{\boldsymbol{b}}, w_{\boldsymbol{b}})$ is the weighted polyhedral complex associated with a simple valuation for at most one unit of any good, and apply Theorem A.2. This then extends to the case $w(\boldsymbol{b}) = -1$ because changing the sign of the weights of all facets does not affect balancing. Then, $(\Pi_{\mathcal{B}}, w_{\mathcal{B}})$ is also balanced as it is the complex corresponding to the union of balanced LIPs.

Next, we see that all weights of $(\Pi_{\mathcal{B}}, w_{\mathcal{B}})$ are nonnegative. Let F be a facet of $\Pi_{\mathcal{B}}$ and fix a point $\boldsymbol{p}$ in the relative interior of F. Then, because the normal vector to F is a strong-substitutes vector and by Proposition A.1, every bid is either

nonmarginal at $\boldsymbol{p}$ or marginal between the same two distinct goods, $i, i' \in [n]_0$. Let $\mathcal{B}_{ii'}$ denote the bids that are marginal between i, i' at $\boldsymbol{p}$. Note that we have $F \subseteq \mathcal{L}_{\boldsymbol{b}}$ if and only if $\boldsymbol{b}$ is marginal on i, i', so $w_{\mathcal{B}}(F) = \sum_{\boldsymbol{b} \in \mathcal{B}_{ii'}} w(\boldsymbol{b}) \geq 0$ (by assumption (2)).

$(\Pi_{\mathcal{B}}, w_{\mathcal{B}})$ may have zero-weighted facets: let the set of these be $\Pi^0_{\mathcal{B}}$ and write $\Pi^+_{\mathcal{B}} := \Pi_{\mathcal{B}} \setminus \Pi^0_{\mathcal{B}}$. Then, $\Pi^+_{\mathcal{B}}$ inherits from $\Pi_{\mathcal{B}}$ the structure of a polyhedral complex. Moreover, if we write $w^+_{\mathcal{B}}$ for the restriction of $w_{\mathcal{B}}$ to the facets of $\Pi^+_{\mathcal{B}}$, then $(\Pi^+_{\mathcal{B}}, w^+_{\mathcal{B}})$ is balanced: removing 0-weighted facets does not affect this criterion.

So we write $\mathcal{L}^+_{\mathcal{B}}$ for the union of the cells in $\Pi^+_{\mathcal{B}}$. Let $\overline{\boldsymbol{p}}$ be a price vector whose components are strictly larger than the components of any bid so that $D_{\mathcal{B}}(\overline{\boldsymbol{p}}) = \boldsymbol{0}$. By Theorem A.2, there exists a unique concave-extensible valuation u such that $\mathcal{L}_u = \mathcal{L}^+_{\mathcal{B}}$, $w_u = w^+_{\mathcal{B}}$, $u(\boldsymbol{0}) = 0$ and $D_u(\overline{\boldsymbol{p}}) = \boldsymbol{0}$. We observe that all facets of $\Pi_{\mathcal{B}}$ and, hence, of $\mathcal{L}_u$ are normal to a strong-substitutes vector, so u is a strong-substitutes valuation by Theorem A.1.

By Proposition A.1, the facets and weights of $\mathcal{L}_u$ define all changes in demand between UDRs associated with valuation u. If $w(\boldsymbol{b})$ is positive, then $\mathcal{L}_{\boldsymbol{b}}$ has the property that facets and weights define all changes in demand between UDRs (see also Baldwin and Klemperer [2, section 2.2]), and it is easy to see that the same follows for negative-weighted bids and, thus, extends by aggregation to our positive-weighted polyhedral complex $(\Pi^+_{\mathcal{B}}, w^+_{\mathcal{B}})$. So $D_u(\overline{\boldsymbol{p}}) = D_{\mathcal{B}}(\overline{\boldsymbol{p}})$ implies $D_u(\boldsymbol{p}) = D_{\mathcal{B}}(\boldsymbol{p})$ for all UDR prices $\boldsymbol{p} \in \mathbb{R}^n$ and, hence, for all prices because u is concave-extensible and $D_{\mathcal{B}}(\boldsymbol{p})$ is defined to be discrete convex.

We now show that $f_{\mathcal{B}} = f_u$. For any price $\boldsymbol{p}$ at which demand is unique for every bid $\boldsymbol{b} \in \mathcal{B}$, let $\boldsymbol{x}$ be the bundle demanded and let $(i(\boldsymbol{b}))_{\boldsymbol{b} \in \mathcal{B}}$ be the list of goods allocated to the bids. That is, each bid $\boldsymbol{b} \in \mathcal{B}$ uniquely demands $i(\boldsymbol{b})$ and $\boldsymbol{x} = \sum_{\boldsymbol{b} \in \mathcal{B}} w(\boldsymbol{b}) \boldsymbol{e}^{i(\boldsymbol{b})}$. Then, we claim that

$$u(\boldsymbol{x}) = \sum_{\boldsymbol{b} \in \mathcal{B}} w(\boldsymbol{b}) b_{i(\boldsymbol{b})}. \tag{A.1}$$

This then immediately implies $f_{\mathcal{B}}(\boldsymbol{p}) = f_u(\boldsymbol{p})$ for any price $\boldsymbol{p} \in \mathbb{R}^n$ at which demand is unique for every bid because

$$f_{\mathcal{B}}(\boldsymbol{p}) = \sum_{\boldsymbol{b} \in \mathcal{B}} w(\boldsymbol{b}) \max_{i \in [n]_0} (b_i - p_i) = \sum_{\boldsymbol{b} \in \mathcal{B}} w(\boldsymbol{b}) (b_{i(\boldsymbol{b})} - p_{i(\boldsymbol{b})}) = u(\boldsymbol{x}) - \boldsymbol{x} \cdot \boldsymbol{p} = f_u(\boldsymbol{p}).$$

But these prices are dense in $\mathbb{R}^n$, so it follows by continuity of these piecewise-linear functions that $f_{\mathcal{B}}(\boldsymbol{p}) = f_u(\boldsymbol{p})$ for all $\boldsymbol{p} \in \mathbb{R}^n$.

To see that (A.1) holds, first observe that it holds for the demand and allocation at price $\overline{\boldsymbol{p}}$: here, by definition $i(\boldsymbol{b}) = 0$ for all $\boldsymbol{b} \in \mathcal{B}$, so $\boldsymbol{x} = \boldsymbol{0}$, and $u(\boldsymbol{0}) = 0$.

Next, consider price vectors $\boldsymbol{p}', \boldsymbol{p}''$ at which bundles $\boldsymbol{x}', \boldsymbol{x}''$ are uniquely demanded via unique allocations to bids $(i'(\boldsymbol{b}))_{\boldsymbol{b} \in \mathcal{B}}$ and $(i''(\boldsymbol{b}))_{\boldsymbol{b} \in \mathcal{B}}$, respectively. Suppose moreover that there exists a price $\boldsymbol{p}$ satisfying $\boldsymbol{x}', \boldsymbol{x}'' \in D_{\mathcal{B}}(\boldsymbol{p}) = D_u(\boldsymbol{p})$ with $i'(\boldsymbol{b}), i''(\boldsymbol{b}) \in \arg\max_{i \in [n]_0} (b_i - p_i)$ for all $\boldsymbol{b} \in \mathcal{B}$: this holds if $\boldsymbol{p}'$ and $\boldsymbol{p}''$ are in the same or neighboring connected components of the complement of $\Pi_{\mathcal{B}}$ in $\mathbb{R}^n$. Then, we have

$$u(\boldsymbol{x}') - \boldsymbol{p} \cdot \boldsymbol{x}' = u(\boldsymbol{x}'') - \boldsymbol{p} \cdot \boldsymbol{x}'' \tag{A.2}$$

and

$$f_{\mathcal{B}}(\boldsymbol{p}) = \sum_{\boldsymbol{b} \in \mathcal{B}} w(\boldsymbol{b}) (b_{i'(\boldsymbol{b})} - p_{i'(\boldsymbol{b})}) = \sum_{\boldsymbol{b} \in \mathcal{B}} w(\boldsymbol{b}) (b_{i''(\boldsymbol{b})} - p_{i''(\boldsymbol{b})}). \tag{A.3}$$

Suppose (A.1) holds for bundle $\boldsymbol{x}'$ and $(i'(\boldsymbol{b}))_{\boldsymbol{b} \in \mathcal{B}}$. Then, (A.2) and (A.3) imply that (A.1) holds for $\boldsymbol{x}''$ and $(i''(\boldsymbol{b}))_{\boldsymbol{b} \in \mathcal{B}}$. But now we can use the fact that (A.1) holds for the demand and allocation at price $\overline{\boldsymbol{p}}$ to infer that (A.1) for the demand and allocation at any price at which demand is unique by induction. □

A.2. Proof of Proposition 1

We make use of the following technical observation to prove Proposition 1.

Observation A.1. Let $f : [0,1] \to \mathbb{R}$ be a continuous, piecewise-linear function. If its slope is nondecreasing as we move from 0 to 1, f is convex. This implies that, if f is not convex, there exists a point λ at which two linear segments meet and the slope decreases.

Proof of Proposition 1. Recall that $f_{\mathcal{B}}$ restricted to P is convex if and only if

$$f_{\mathcal{B}}(\lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{q}) \leq \lambda f_{\mathcal{B}}(\boldsymbol{p}) + (1-\lambda) f_{\mathcal{B}}(\boldsymbol{q}), \quad \forall \boldsymbol{p}, \boldsymbol{q} \in P \text{ and } \lambda \in [0,1]. \tag{A.4}$$

Suppose that $\boldsymbol{p}$ and $\boldsymbol{q}$ are two nonmarginal prices in neighboring UDRs such that, for some $\lambda \in (0,1)$, the point $\boldsymbol{r} = \lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{q}$ lies on the interior of a facet separating the UDRs. This implies two possibilities for each bid $\boldsymbol{b} \in \mathcal{B}$. Either $\boldsymbol{b}$ nonmarginally demands the same good at $\boldsymbol{p}$ and $\boldsymbol{q}$ (and, thus, also at $\boldsymbol{r}$), or $\boldsymbol{b}$ nonmarginally demands i at $\boldsymbol{p}$ and i' at $\boldsymbol{q}$ and is marginal (only) on i and i' at $\boldsymbol{r}$. Moreover, all bids of the latter kind are marginal on the same two goods.

Let $\mathcal{B}_{ii'}$ denote the bids that demand the set $\{i, i'\}$ at $\boldsymbol{r}$. We show that $f_{\mathcal{B}}$ satisfies (A.4) for $\boldsymbol{p}, \boldsymbol{q}$ and any $\lambda \in [0,1]$ if and only if the weights of the bids marginal on i and i' sum to a nonnegative number, that is, if $\sum_{\boldsymbol{b} \in \mathcal{B}_{ii'}} w(\boldsymbol{b}) \geq 0$. By construction,

we have

$$f_{\mathcal{B}}(\boldsymbol{p}) = \sum_{\boldsymbol{b}\in\mathcal{B}_{ii'}} w(\boldsymbol{b})(b_i - p_i) + \sum_{\boldsymbol{b}\notin\mathcal{B}_{ii'}} w(\boldsymbol{b}) \max_{j\in[n]_0}(b_j - p_j),$$
$$f_{\mathcal{B}}(\boldsymbol{q}) = \sum_{\boldsymbol{b}\in\mathcal{B}_{ii'}} w(\boldsymbol{b})(b_{i'} - q_{i'}) + \sum_{\boldsymbol{b}\notin\mathcal{B}_{ii'}} w(\boldsymbol{b}) \max_{j\in[n]_0}(b_j - q_j),$$
$$f_{\mathcal{B}}(\boldsymbol{r}) = \sum_{\boldsymbol{b}\in\mathcal{B}_{ii'}} w(\boldsymbol{b})(b_i - r_i) + \sum_{\boldsymbol{b}\notin\mathcal{B}_{ii'}} w(\boldsymbol{b}) \max_{j\in[n]_0}(b_j - r_j).$$

Hence, the inequality

$$f_{\mathcal{B}}(\boldsymbol{r}) \le \lambda f_{\mathcal{B}}(\boldsymbol{p}) + (1-\lambda) f_{\mathcal{B}}(\boldsymbol{q}) \tag{A.5}$$

holds if and only if

$$\sum_{\boldsymbol{b}\in\mathcal{B}_{ii'}} w(\boldsymbol{b})[(b_i - q_i) - (b_{i'} - q_{i'})] \le 0. \tag{A.6}$$

Note that the term $(b_i - q_i) - (b_{i'} - q_{i'})$ is strictly negative as $\boldsymbol{b}$ demands i' and not i at $\boldsymbol{q}$. Moreover, this term is the same for all bids $\boldsymbol{b} \in \mathcal{B}_{ii'}$ as i and i' are both demanded at $\boldsymbol{r}$. This implies by (A.6) that $\sum_{\boldsymbol{b}\in\mathcal{B}_{ij}} w(\boldsymbol{b}) \ge 0$ if and only if (A.5) holds.

Now, we prove our main statement. Suppose $f_{\mathcal{B}}$ restricted to P is not convex. Then, there exist $\boldsymbol{p},\boldsymbol{q} \in P$, and $\lambda \in [0,1]$ that violate (A.4). Because of continuity of $f_{\mathcal{B}}$, we can perturb $\boldsymbol{p}$ and $\boldsymbol{q}$ slightly so that (A.4) is still violated, $\boldsymbol{p},\boldsymbol{q}$ are nonmarginal, and the line segment $[\boldsymbol{p},\boldsymbol{q}]$ crosses only interiors of facets of the LIP $\mathcal{L}_{\mathcal{B}}$. We can assume without loss of generality that $\boldsymbol{p}$ and $\boldsymbol{q}$ are in neighboring UDRs; otherwise, we can apply Observation A.1 to find two nonmarginal prices on the interior of the line segment for which (A.4) fails. Thus, we can apply the preceding to see the implication $2 \Rightarrow 1$.

For the converse ($1 \Rightarrow 2$), suppose there exist $\boldsymbol{r}$ and $i,i' \in [n]_0$ such that the weights of bids marginal on i,i' at $\boldsymbol{r}$ sum to a negative number. Without loss of generality, we can assume that all marginal bids at $\boldsymbol{r}$ are marginal only on goods i,i' by subtracting $\delta\boldsymbol{e}^{\{i,i'\}}$ from $\boldsymbol{r}$ for some infinitesimal positive value of δ if necessary. Hence, $\boldsymbol{r}$ lies in the interior of a facet F, and for small enough $\varepsilon > 0$, the points $\boldsymbol{p} = \boldsymbol{r} + \varepsilon\boldsymbol{e}^i - \varepsilon\boldsymbol{e}^{i'}$ and $\boldsymbol{q} = \boldsymbol{r} - \varepsilon\boldsymbol{e}^i + \varepsilon\boldsymbol{e}^{i'}$ lie in P and in neighboring UDRs separated by F. This implies that $\boldsymbol{p}$ and $\boldsymbol{q}$ with $\lambda = 1/2$ violate (A.4), and we are done. □

A.3. Proof of Lemma 1

Proof of Lemma 1. We show that $f_{\mathcal{B}}$ is convex on $B(\boldsymbol{p},\varepsilon/2)$ by verifying that it satisfies midpoint convexity. Note that $f_{\mathcal{B}}$ is linear on the line segment connecting $\boldsymbol{p}$ to $\boldsymbol{q}$ for any $\boldsymbol{q} \in B(\boldsymbol{p},\varepsilon/2)$. Indeed, $f_{\mathcal{B}}$ is the sum of terms $w(\boldsymbol{b})\max_{i\in[n]_0}(b_i - p_i)$, and it suffices to show that each term is linear. Hence, fix $\boldsymbol{b}$ and define $h(\boldsymbol{p}) = w(\boldsymbol{b})\max_{i\in[n]_0}(b_i - p_i)$ as well as $\boldsymbol{r} = \theta\boldsymbol{p} + (1-\theta)\boldsymbol{q}$. By Observation 1, the goods demanded by $\boldsymbol{b}$ at $\boldsymbol{q},\boldsymbol{r}$ and $\boldsymbol{p}$ satisfy $I_q \subseteq I_r \subseteq I_p$. Hence, for any $i^* \in I_q$, we have $h(\boldsymbol{p}) = w(\boldsymbol{b})(b_{i^*} - p_{i^*}) = \theta h(\boldsymbol{p}) + (1-\theta)h(\boldsymbol{q})$. Now, fix $\boldsymbol{q},\boldsymbol{q}' \in B(\boldsymbol{p},\varepsilon/2)$ and choose $\theta > 0$ so that $\boldsymbol{r} = \theta\boldsymbol{p} + (1-\theta)\boldsymbol{q}$, $\boldsymbol{r}' = \theta\boldsymbol{p} + (1-\theta)\boldsymbol{q}'$, and $(\boldsymbol{r}+\boldsymbol{r}')/2$ are in $B(\boldsymbol{p},\delta)$. As $f_{\mathcal{B}}$ is convex on $B(\boldsymbol{p},\delta)$, by assumption, we have $f((\boldsymbol{r}+\boldsymbol{r}')/2) \le f(\boldsymbol{r})/2 + f(\boldsymbol{r}')/2$. Second, we have $f(\boldsymbol{r}) = \theta f(\boldsymbol{p}) + (1-\theta)f(\boldsymbol{q})$, $f(\boldsymbol{r}') = \theta f(\boldsymbol{p}) + (1-\theta)f(\boldsymbol{q}')$, and $f((\boldsymbol{r}+\boldsymbol{r}')/2) = \theta f(\boldsymbol{p}) + (1-\theta)f((\boldsymbol{q}+\boldsymbol{q}')/2)$ because of the linearity of $f_{\mathcal{B}}$ on the line segments connecting $\boldsymbol{p}$ to $\boldsymbol{q},\boldsymbol{q}'$ and $(\boldsymbol{q}+\boldsymbol{q}')/2$. This implies midpoint convexity, $f((\boldsymbol{q}+\boldsymbol{q}')/2) \le f(\boldsymbol{q})/2 + f(\boldsymbol{q}')/2$. □

A.4. Proof of Proposition 2

In order to prove Proposition 2, we first prove in Lemma A.1 that the demand of a bid list that is locally valid around integral prices coincides, in the local neighborhood of these prices, with the demand of some strong-substitutes valuation (in analogy to the third statement in Theorem 1). Working directly with the valuations that correspond locally to our bid lists, we then establish two technical lemmas (Lemmas A.2 and A.3) that illustrate the robustness of demand when we perturb valuations by some small value. We recall that a vector is strong substitutes if it has at most one +1 entry, at most one −1 entry, and no other nonzero entries. In Lemma A.3 and the proof of Proposition 2, we refer to the aggregate valuation of two valuations. The aggregation of two bid lists is defined as the concatenation of the two individual lists. For any two strong-substitutes valuations u^1 and u^2 associated with valid bid lists $\mathcal{B}^1$ and $\mathcal{B}^2$, we define the aggregate valuation $u^{1,2}$ as the strong-substitutes valuation associated with the aggregation of the two bid lists $\mathcal{B}^1$ and $\mathcal{B}^2$; note that the existence of $u^{1,2}$ is guaranteed by Theorem 1.[14]

Lemma A.1. *Let $\mathcal{B}$ be an integral bid list that is ε-valid at integral prices $\boldsymbol{p}$ with $\varepsilon < 1/2$. There exists a strong-substitutes valuation u such that $D_{\mathcal{B}}(\boldsymbol{q}) = D_u(\boldsymbol{q})$ for all $\boldsymbol{q} \in B(\boldsymbol{p};\varepsilon)$.*

Proof of Lemma A.1. Let $\mathcal{H}$ be the collection of hyperplanes obtained by taking the affine hull of all facets in the polyhedral complex $\Pi_{\mathcal{B}}$ of $\mathcal{B}$ that intersect $B(\boldsymbol{p};\varepsilon)$. Let $H \in \mathcal{H}$ be a hyperplane obtained from facet F of $\Pi_{\mathcal{B}}$. As F is normal to a strong-substitutes vector and the bid entries are integral, we can write $H = \{\boldsymbol{q} \in \mathbb{R}^n \mid q_i - q_j = c\}$ for some $i,j \in [n]_0$ and some integer c. For any $\boldsymbol{q} \in B(\boldsymbol{p};\varepsilon) \cap H$, we have $p_i - p_j - 2\varepsilon < q_i - q_j < p_i - p_j + 2\varepsilon$. Because of integrality of c and $\varepsilon < 1/2$, we have $q_i - q_j = p_i - q_j = c$. Hence, every $H \in \mathcal{H}$ contains $\boldsymbol{p}$.

We now construct a second rational polyhedral complex (Π, w) as follows. Take the union of all hyperplanes in $\mathcal{H}$ and decompose this union into a polyhedral complex to obtain Π. The cells of this complex are intersections of hyperplanes in $\mathcal{H}$. Every cell (and so every facet and $(n-2)$-dimensional cell) of Π contains $\boldsymbol{p}$ as $\boldsymbol{p}$ is contained in every $H \in \mathcal{H}$. So, in particular, $F \cap P \neq \emptyset$ for every facet F of Π. Moreover, for each facet F of Π, either $F \cap P \subseteq F' \cap P$ for some facet F' of $\Pi_{\mathcal{B}}$ or $F \cap P$ has at most $(n-2)$-dimensional intersection with the facets of $\Pi_{\mathcal{B}}$; this holds because the facets of $\Pi_{\mathcal{B}}$ meet facets with distinct affine spans at every point on their boundaries. This allows us to define the weight function w as follows. For any facet F of Π, we let $w(F) = w_{\mathcal{B}}(F')$ if $F \cap P \subseteq F' \cap P$ for some facet F' of $\Pi_{\mathcal{B}}$, and we let $w(F) = 0$ otherwise. Note that (Π, w) is balanced within P because $(\Pi_{\mathcal{B}}, w_{\mathcal{B}})$ is balanced (see also the proof of Theorem 1): dividing a facet of a balanced polyhedral complex into two equally weighted facets and adding zero-weighted facets does not affect balancing. But every $(n-2)$-cell of Π and every facet of Π containing this $(n-2)$-cell has nonzero intersection with P. So (Π, w) is balanced.

The facets of $\Pi_{\mathcal{B}}$ that intersect with P have nonnegative weight because of the local validity of $\mathcal{B}$ at $\boldsymbol{p}$ (by the same argument as in the proof of Theorem 1). Hence, the weight function w is nonnegative by construction. Now, let (Π^+, w^+) be obtained from (Π, w) by removing all zero-weighted facets and let $\mathcal{L}^+$ be the union of the facets of Π^+. Note that $\mathcal{L}_{\mathcal{B}} \cap P = \mathcal{L}^+ \cap P$, and if F' is a facet of $\mathcal{L}_{\mathcal{B}}$ and F^+ is a facet of $\mathcal{L}^+$ such that $\dim(F \cap F^+ \cap P) = n-1$, then $w(F') = w^+(F^+)$. Fix some prices $\boldsymbol{q} \in P$, $\boldsymbol{q} \notin \mathcal{L}^+$ and let $\boldsymbol{x}$ be the bundle uniquely demanded by $\mathcal{B}$ at $\boldsymbol{q}$. By Theorem A.2, there exists a valuation function u such that $D_u(\boldsymbol{q}) = \{\boldsymbol{x}\}$, $\mathcal{L}_u = \mathcal{L}^+$ and $w_u = w^+$. Using the same argument as in the proof of Theorem 1, we have that the normal vectors and weights of the facets of $\mathcal{L}_{\mathcal{B}}$ fully specify how demand of $\mathcal{B}$ changes in P. Hence, by Proposition A.1, it follows that $D_{\mathcal{B}}(\boldsymbol{q}') = D_u(\boldsymbol{q}')$ for all $\boldsymbol{q}' \in P$. Moreover, as the facets of $\mathcal{L}_u$ are normal to strong-substitutes vectors, the valuation u is strong-substitutes by Theorem A.1. □

Lemma A.2. *Suppose that $G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}$ is an invertible $n \times n$ matrix with strong-substitutes rows. Moreover, let H_i denote the ith column of the matrix $H := \begin{pmatrix} 0 \\ G_2 \end{pmatrix}$. Then, for any $i \in [n]$, we have either (i) $G^{-1}H_i = \boldsymbol{e}^S$ for some $S \subseteq [n]$ with $i \in S$ or (ii) $G^{-1}H_i = -\boldsymbol{e}^S$ for some $S \subseteq [n]$ with $i \notin S$.*

Proof. Suppose G is a lower triangular matrix with diagonal values $g_{ii} = 1$ and $g_{ij} \in \{0, -1\}$ for all $i > j$. This is without loss of generality as we can permute columns and multiply by a diagonal matrix U with entries ± 1 on its diagonals otherwise; in this case, replace G by PUG (where P denotes the permutation matrix) below and adjust the proof slightly. Note that the rows of PUG remain strong substitutes.

In order to prove our lemma, we associate G with a directed graph $\mathcal{D}$ on vertices $[n]_0$ as follows. As G is invertible and the rows are strong substitutes, each row takes the form $\boldsymbol{e}^j - \boldsymbol{e}^i$ for some distinct $i, j \in [n]_0$, where we define $\boldsymbol{e}^0 = \boldsymbol{0}$. Thus, for each row $\boldsymbol{e}^j - \boldsymbol{e}^i$ we add an arc from i to j.

We establish some properties of $\mathcal{D}$. First, $\mathcal{D}$ is an arborescence rooted at 0; that is, it is a directed graph in which there is exactly one path from 0 to any other vertex. This can be seen by induction on $k \in [n]$ if we define G_{kk} as the submatrix of G consisting of the first k rows and columns and let $\mathcal{D}_{kk}$ denote its graph. It is immediate that $\mathcal{D}_{11}$ is an arborescence. Second, consider $\mathcal{D}_{kk}$ and assume the claim holds for $k-1$. The last row (and column) of G_{kk} adds a single arc from a good $j < k$ to k, so $\mathcal{D}_{kk}$ remains an arborescence.

Second, we can express G as $G = I_n - L$, where I_n is the identity matrix and L is a strictly lower triangular matrix. Moreover, L admits the intuitive interpretation of being the transpose of the adjacency matrix of $\mathcal{D}$. (To see this, note that $\boldsymbol{e}^j - \boldsymbol{e}^i$ is the jth row in G, so $L_{ji} = 1$ if and only if there exists an arc from i to j.) In the following, let L^k be the kth power of L (with $L^0 = I_n$). We recall the well-known result that L^k_{ji} denotes the number of directed walks of length k from vertex i to j. In particular, as $\mathcal{D}$ is a tree, all walks are paths. This implies that, for all $i, j \in [n]$, we have $L^k_{ij} \in \{0, 1\}$ and $L^k_{ij} = 1$ for at most one value of k. Moreover, $L^n = \boldsymbol{0}$ as no path can have length n. It follows that the inverse of G is

$$G^{-1} = \sum_{k=0}^{n-1} L^k, \tag{A.7}$$

and G^{-1} has only entries $\{0, 1\}$.

Fix $i \in [n]$ and let J_i be any subset of the in-neighbors of i. (The in-neighbors of vertex i in $\mathcal{D}$ are the vertices for which there exists an arc from i to j.) We claim that, for $\boldsymbol{h} = -\sum_{j \in J_i} \boldsymbol{e}^j$, there exists a subset S such that $i \in S$ and $G^{-1}\boldsymbol{h} = -\boldsymbol{e}^S$ and for $\boldsymbol{h} = \boldsymbol{e}^i - \sum_{j \in J_i} \boldsymbol{e}^j$, there exists a subset S such that $i \in S$ and $G^1\boldsymbol{h} = \boldsymbol{e}^S$. This claim immediately implies the main statement of Lemma A.2 as H_i takes the form $H_i = -\sum_{j \in J_i} \boldsymbol{e}^j$ or $H_i = \boldsymbol{e}^i - \sum_{j \in J_i} \boldsymbol{e}^j$ for some J_i.

It remains to prove our claim. Note first that $J_i \subseteq \{i+1, \ldots, n\}$. For any $j \in [n]_0$, let $S(j)$ denote the vertices reachable from vertex j in $\mathcal{D}$. As $L^k_{jl} = 1$ if there exists a path of length k from vertex l to j and $L^k_{jl} = 0$ otherwise, (A.7) implies that $G^{-1}\boldsymbol{e}^i = \boldsymbol{e}^{S(j)}$ is the indicator vector denoting, for each $l \in [n]$, whether l is reachable from j (along a path of arbitrary length). Next, observe that J_i contains endpoints of arcs from i, so the arborescence property of $\mathcal{D}$ implies that the sets $S(j)$ for $j \in J_i$ are pairwise disjoint. Moreover, we see that $S(j) \subseteq S(i)$ as there is an arc from i to every $j \in J_i$. This allows us to say the following. If $\boldsymbol{h} = \sum_{j \in J_i} \boldsymbol{e}^j$, then we have $G^{-1}\boldsymbol{h} = -\sum_{j \in J_i} \boldsymbol{e}^{S(j)} = \boldsymbol{e}^S$, where we define $S := \cup_{j \in J_i} S(j)$, and we note that

$i \notin S$. On the other hand, if $\boldsymbol{h} = \boldsymbol{e}^i - \sum_{j \in J_i} \boldsymbol{e}^j$, then we have $G^{-1}\boldsymbol{h} = \boldsymbol{e}^{S(i)} - \sum_{j \in J_i} \boldsymbol{e}^{S(j)} = \boldsymbol{e}^S$, where we define $S := S(i) \setminus \cup_{j \in J_i} S(j)$, and we note that $i \in S$. The last equality holds as $S(j) \subseteq S(i)$ for all $j \in J_i$. □

Observation A.2. Suppose that G is an $n' \times n$ matrix with strong-substitutes rows and $S \subseteq [n]$. Then, $G\boldsymbol{e}^S \in \{-1, 0, 1\}^n$, and in particular, $G\boldsymbol{e}^S \geq -\mathbf{1}$.

Lemma A.3. *Suppose we have two strong-substitutes integer-valued valuations, u^1 and u^2, and let $u^{1,2}$ denote the aggregate valuation. Fix $i \in [n]$ and $\varepsilon \in (0,1)$, and write $u^{2[\varepsilon,i]}$ for the valuation given by $u^{2[\varepsilon,i]}(\boldsymbol{x}) = u^2(\boldsymbol{x}) + \varepsilon x_i$. For any $\boldsymbol{r}^1, \boldsymbol{r}^2 \in \mathbb{Z}^n$, suppose there exists a price $\boldsymbol{p}^*$ such that $\boldsymbol{r}^1 \in D_{u^1}(\boldsymbol{p}^*)$, and $\boldsymbol{r}^2 \in D_{u^{2[\varepsilon,i]}}(\boldsymbol{p}^*)$. Then, there exists a price $\boldsymbol{p}^0$ such that $\boldsymbol{r}^1 \in D_{u^1}(\boldsymbol{p}^0)$ and $\boldsymbol{r}^2 \in D_{u^2}(\boldsymbol{p}^0)$. Moreover, for any price $\boldsymbol{p}$ such that $\boldsymbol{r} := \boldsymbol{r}^1 + \boldsymbol{r}^2 \in D_{u^{1,2}}(\boldsymbol{p})$, we have $\boldsymbol{r}^1 \in D_{u^1}(\boldsymbol{p})$ and $\boldsymbol{r}^2 \in D_{u^2}(\boldsymbol{p})$.*

Proof. Fix bundles $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$ that are, by assumption, demanded at $\boldsymbol{p}^*$ under u^1 and $u^{2[\varepsilon,i]}$. For $j \in \{1,2\}$, we define $P^j := \{\boldsymbol{p} \geq 0 \,|\, \boldsymbol{r}^j \in D_{u^j}(\boldsymbol{p})\}$, the set of prices at which $\boldsymbol{r}^j$ is demanded under u^j. It follows from the definition of $u^{2[\varepsilon,i]}$ that the set $P^{2[\varepsilon,i]}$ of prices at which $\boldsymbol{r}^2$ is demanded by $\boldsymbol{p}$ under $u^{2[\varepsilon,i]}$ is equal to $P^2 + \varepsilon\boldsymbol{e}^i$. Moreover, by assumption, we have $\boldsymbol{p}^* \in P^1 \cap P^{2[\varepsilon,i]}$, so this intersection is nonempty.

As P^1 and P^2 are polyhedra in $\mathbb{R}^n$ (cf. Baldwin and Klemperer [2, proposition 2.7]), we can describe each region P^j by the system of linear inequalities

$$H^j\boldsymbol{p} \geq A^j, \tag{A.8}$$

where H^j is an $a_j \times n$ matrix for some $a_j \in \mathbb{Z}_+$ and $A^j \in \mathbb{R}^{a_j}$. Moreover, Baldwin and Klemperer [2, proposition 3.10] tells us that we can choose H^j such that each row of H^j is a strong-substitutes vector. Additionally, as valuation u^j is integer-valued, we have $A^j \in \mathbb{Z}^{a_j}$. It follows that $P^{2[\varepsilon,i]} = P^2 + \varepsilon\boldsymbol{e}^i$ is defined by

$$H^2(\boldsymbol{p} - \varepsilon\boldsymbol{e}^i) \geq A^2 \Leftrightarrow H^2\boldsymbol{p} \geq A^2 + \varepsilon H^2_i, \tag{A.9}$$

where H^2_i is the ith column of H^2. This allows us to express the intersection $P^1 \cap P^{2[\varepsilon,i]}$ as

$$H\boldsymbol{p} \geq A + \varepsilon\hat{H}_i, \tag{A.10}$$

where we define $H = \begin{pmatrix} H^1 \\ H^2 \end{pmatrix}$, $A = \begin{pmatrix} A^1 \\ A^2 \end{pmatrix}$, and $\hat{H} = \begin{pmatrix} 0 \\ H^2 \end{pmatrix}$.

Now, we argue similarly to Schrijver [35, theorem 19.1]. We know that $P^1 \cap P^{2[\varepsilon,i]} \neq \emptyset$. Let $F = \{\boldsymbol{p} \geq 0 \,|\, H'\boldsymbol{p} = A' + \varepsilon\hat{H}'_i\}$ be a minimal face of this intersection, in which $H'\boldsymbol{p} \geq A' + \varepsilon\hat{H}'_i$ is a subsystem of (A.10) with linearly independent rows. Without loss of generality, we can write $H' = (G \;\; \tilde{G})$, where G is invertible (permuting the columns of H' if necessary, which is equivalent to relabeling goods). Moreover, we can choose this permutation such that the ith coordinate remains within the first square matrix; that is, if G is $n' \times n'$, then $i \leq n'$. In addition, G inherits from H^1, H^2 the property that its rows are strong-substitutes vectors. Now, define the two price vectors

$$\boldsymbol{p}^\varepsilon := \begin{pmatrix} G^{-1}(A' + \varepsilon\hat{H}'_i) \\ 0 \end{pmatrix} \text{ and } \quad \boldsymbol{p}^0 := \begin{pmatrix} G^{-1}A' \\ 0 \end{pmatrix}.$$

Note that $\boldsymbol{p}^0$ is an integral vector because A is integral and G is unimodular. Moreover, by Lemma A.2, we know that either (i) $G^{-1}\hat{H}'_i = \boldsymbol{e}^S$ for some $S \subseteq [n']$ with $i \in S$ or (ii) $G^{-1}\hat{H}'_i = -\boldsymbol{e}^S$ for some $S \subseteq [n']$ with $i \notin S$ (here, n' is the dimension of G). In either case, $\boldsymbol{p}^\varepsilon - \varepsilon\boldsymbol{e}^i \in \{\boldsymbol{p}^0 \pm \varepsilon\boldsymbol{e}^{S'}\}$ for some $S' \subseteq [n]$. We also see immediately that $\boldsymbol{p}^\varepsilon$ is a vector in F and, hence, lies in $P^1 \cap P^{2[\varepsilon,i]}$, so $\boldsymbol{p}^\varepsilon$ satisfies (A.10). We now use this knowledge to show that $\boldsymbol{p}^0 \in P^1 \cap P^2$.

As $\boldsymbol{p}^\varepsilon \in P^1$, it satisfies (A.8) for $j = 1$. Note also that the rows of H^1 are strong substitutes and $(\boldsymbol{p}^0 - \boldsymbol{p}^\varepsilon) \in \{\pm\varepsilon\boldsymbol{e}^S\}$ for some $S \subseteq [n]$, so (A.8) and Observation A.2 imply

$$H^1\boldsymbol{p}^0 = H^1\boldsymbol{p}^\varepsilon + H^1(\boldsymbol{p}^0 - \boldsymbol{p}^\varepsilon) \geq A^1 - \varepsilon\mathbf{1}.$$

The integrality of H^1, $\boldsymbol{p}^0$, and A^1, together with $\varepsilon \in (0,1)$, imply $H^1\boldsymbol{p}^0 \geq A^1$. Similarly, we argue that $H^2\boldsymbol{p}^0 \geq A^2$. Indeed, $\boldsymbol{p}^varepsilon$ satisfies (A.9), and we have $(\boldsymbol{p}^0 - \boldsymbol{p}^\varepsilon + \varepsilon\boldsymbol{e}^i) \in \pm\varepsilon\boldsymbol{e}^S$ for some $S \subseteq [n]$, so (A.9) and Observation A.2 imply

$$H^2\boldsymbol{p}^0 = H^2(\boldsymbol{p}^\varepsilon - \varepsilon\boldsymbol{e}^i) + H^2(\boldsymbol{p}^0 - \boldsymbol{p}^\varepsilon + \varepsilon\boldsymbol{e}^i) \geq A^2 - \varepsilon\mathbf{1} \geq A^2.$$

The last inequality again follows because of the integrality of $H^2, \boldsymbol{p}^0$ and A^2. Thus, we have established that $\boldsymbol{p}^0 \in P^1 \cap P^2$, which is equivalent to $\boldsymbol{r}^j \in D_{u^j}(\boldsymbol{p}^0)$ for $j = 1, 2$, our first claim.

Now, let $\boldsymbol{r} := \boldsymbol{r}^1 + \boldsymbol{r}^2$ and let $\boldsymbol{p}$ be a price at which $\boldsymbol{r}$ is aggregately demanded, so $\boldsymbol{r} \in D_{u^{1,2}}(\boldsymbol{p})$. Suppose $\boldsymbol{r} = \boldsymbol{s}^1 + \boldsymbol{s}^2$ is a potentially different equilibrium allocation of bundles across the two agents so that $\boldsymbol{s}^j \in D_{u^j}$ holds. By Mas-Colell et al. [21, proposition 2.F.1], we know that $(\boldsymbol{s}^j - \boldsymbol{r}^j) \cdot (\boldsymbol{p} - \boldsymbol{p}^0) \leq 0$ for $j = 1, 2$ with equality holding only if the agent is indifferent between both bundles at both prices. Hence, adding across $j = 1, 2$, we obtain $((\boldsymbol{s}^1 + \boldsymbol{s}^2) - \boldsymbol{r}) \cdot (\boldsymbol{p} - \boldsymbol{p}^0) \leq 0$ with equality holding only if both agents are indifferent between their two bundles at both prices. But as $\boldsymbol{s}^1 + \boldsymbol{s}^2 = \boldsymbol{r}$, equality holds, so we arrive at $\boldsymbol{r}^j \in D_{u^j}(\boldsymbol{p})$ for $i = 1, 2$. □

Proof of Proposition 2. By Lemma A.1, every bid list that is η-valid at $\boldsymbol{p}$ with $\eta < 1/2$ can be associated with a strong-substitutes integral-valued valuation such that their indirect utility function and demand correspondence coincide in $B(\boldsymbol{p}, \eta)$. The bid lists $\mathcal{B}^j$ and $\mathcal{B} \setminus \mathcal{B}^j$ are 1/2-valid at $\boldsymbol{p}$ by Lemma 1 and the fact that integral bids have a surplus of at least

1 at integral prices. Moreover, the shifted bid list of bidder j and the bid list $\mathcal{B}'$ are $(1/2-|\varepsilon|)$-valid at $\boldsymbol{p}$ (this follows from the definition of local validity). Recalling that $|\epsilon|<1/4$, we fix η with $2|\epsilon|<\eta<1/2$ and let u^2 denote a strong-substitutes integral-valued valuation of bidder j corresponding in $B(\boldsymbol{p},\eta)$ to $\mathcal{B}^j$ and let u^1 denote a (strong-substitutes integral-valued) valuation corresponding in $B(\boldsymbol{p},\eta)$ to the bids $\mathcal{B}\setminus\mathcal{B}^j$ of the other bidders. As before, $u^{1,2}$ denotes the aggregate of these valuations, and it corresponds in $B(\boldsymbol{p},\eta)$ to $\mathcal{B}$. By assumption, we have $\boldsymbol{r}\in D_{u^{1,2}}(\boldsymbol{p})$.

As in Lemma A.3, we write $u^{2[\varepsilon,i]}$ for the valuation given by $u^{2[\varepsilon,i]}(\boldsymbol{x})=u^2(\boldsymbol{x})+\varepsilon x_i$. Note that the polyhedral complex of $u^{2[\varepsilon,i]}$ is the polyhedral complex of u^2 shifted by $\varepsilon<\eta/2$ in direction $\boldsymbol{e}^i$. Hence, in $B(\boldsymbol{p};\eta/2)$, the demand $D_{u^{2[\varepsilon,i]}}$ coincides with the demand of bidder j's bids after shifting, and $D_{u^{1,2[\varepsilon,i]}}$ coincides with $D_{\mathcal{B}'}$ in $B\left(\boldsymbol{p};\eta/2\right)$. Because both u^1 and $u^{2[\varepsilon,i]}$ are strong-substitutes valuations, there exists a price $\boldsymbol{p}^*\in\mathbb{R}^n$ such that $\boldsymbol{r}\in D_{u^{1,2[\varepsilon,i]}}(\boldsymbol{p}^*)$, and thus, there exist $\boldsymbol{r}^1,\boldsymbol{r}^2\in\mathbb{Z}^n$ with $\boldsymbol{r}^1+\boldsymbol{r}^2=\boldsymbol{r}$ such that $\boldsymbol{r}^2\in D_{u^1}(\boldsymbol{p}^*)$ and $\boldsymbol{r}^2\in D_{u^{2[\varepsilon,i]}}(\boldsymbol{p}^*)$. By Lemma A.3, it follows that $\boldsymbol{r}^j\in D_{u^j}(\boldsymbol{p})$ for $j=1,2$.

As in the proof of Lemma A.3, we let $P^j:=\{\boldsymbol{p}'\geq 0\,|\,\boldsymbol{r}^j\in D_{u^j}(\boldsymbol{p}')\}$ for $j\in\{1,2\}$ and observe that $P^{2[\varepsilon,i]}:=\{\boldsymbol{p}'\geq 0\,|\,\boldsymbol{r}^2\in D_{u^{2[\varepsilon,i]}}(\boldsymbol{p}')\}=P^2+\varepsilon\boldsymbol{e}^i$. We know that $\boldsymbol{p}\in P^1\cap P^2$ (so this intersection is nonempty) and $\boldsymbol{p}^*\in P^1\cap P^{2[\varepsilon,i]}$ (so this intersection is also nonempty). We seek $\boldsymbol{p}^\varepsilon\in P^1\cap P^{2[\varepsilon,i]}$ with $\boldsymbol{p}^\varepsilon\in\boldsymbol{p}\pm\varepsilon\boldsymbol{e}^S$ for some $S\subseteq[n]$; this is sufficient to show $\boldsymbol{r}\in D_{u^{1,2[\varepsilon,i]}}(\boldsymbol{p}^\varepsilon)$. Because $|\varepsilon|<\eta/2$, we know $\boldsymbol{p}^\varepsilon\in B(\boldsymbol{p},\eta/2)$, the region in which demand from the valuations coincides with demand from the bids. Our first claim, that $\boldsymbol{r}\in D_{\mathcal{B}'}(\boldsymbol{p}^\varepsilon)$ follows.

Analogous to line (A.8), we express the polyhedron P^j as $H^j\boldsymbol{p}'\geq A^j$ for $j\in\{1,2\}$, where the rows of H^j are strong-substitutes vectors. Therefore, $P^1\cap P^{2[\varepsilon,i]}=P^1\cap(P^2+\varepsilon\boldsymbol{e}^i)$ is defined by the set of inequalities

$$H^1\boldsymbol{p}'\geq A^1 \text{ and } H^2(\boldsymbol{p}'-\varepsilon\boldsymbol{e}^i)\geq A^2. \tag{A.11}$$

Now, for $j=1,2$, let

$$(H^j)'\boldsymbol{p}'\geq(A^j)' \tag{A.12}$$

be the subsystem of $H^j\boldsymbol{p}'\geq A^j$ consisting of all rows that hold with equality at $\boldsymbol{p}$. (If both of these subsystems are empty, then subsequent arguments simply set $\boldsymbol{p}^\varepsilon=\boldsymbol{p}$). Write $H=\begin{pmatrix}(H^1)'\\(H^2)'\end{pmatrix}$, $A=\begin{pmatrix}(A^1)'\\(A^2)'\end{pmatrix}$, and $\hat{H}=\begin{pmatrix}0\\(H^2)'\end{pmatrix}$. Thus, when we stack the corresponding rows of (A.11), we obtain the new system

$$H\boldsymbol{p}'\geq A+\varepsilon\hat{H}_i. \tag{A.13}$$

This defines a superset of $P^1\cap(P^2+\varepsilon)$ because we removed some rows from $H^j\boldsymbol{p}'\geq A^j$ for $j=1,2$. But we see that $P^1\cap(P^2+\varepsilon)$ contains $\boldsymbol{p}^*$, so (A.13) defines a nonempty polytope. So let $F=\{\boldsymbol{p}'\geq 0\,|\,H'\boldsymbol{p}'=A'+\varepsilon\hat{H}_i'\}$ be a minimal face of this polytope, where

$$H'\boldsymbol{p}'\geq A'+\varepsilon\hat{H}_i' \tag{A.14}$$

is a subsystem of (A.13) with linearly independent rows.

Without loss of generality, we can write $H'=(G\ \tilde{G})$, where G is invertible (permuting the columns of H' if necessary, which is equivalent to relabeling goods). Moreover, we can choose this permutation such that the ith coordinate remains within the first square matrix; that is, if G is $n'\times n'$, then assume when we reorder that $i\leq n'$. Correspondingly, write $\boldsymbol{p}'=\begin{pmatrix}\boldsymbol{p}_1'\\\boldsymbol{p}_2'\end{pmatrix}$. In summary, our system (A.13) is now written

$$(G\ \tilde{G})\begin{pmatrix}\boldsymbol{p}_1'\\\boldsymbol{p}_2'\end{pmatrix}\geq A'+\varepsilon\hat{H}_i'. \tag{A.15}$$

By definition of the rows identified at (A.12), we have $(G\ \ \tilde{G})\begin{pmatrix}\boldsymbol{p}_1\\\boldsymbol{p}_2\end{pmatrix}=A'$ and so

$$\boldsymbol{p}_1=G^{-1}A'-G^{-1}\tilde{G}\boldsymbol{p}_2. \tag{A.16}$$

We now define $\boldsymbol{p}^\varepsilon$ as follows:

$$\boldsymbol{p}^\varepsilon:=\begin{pmatrix}G^{-1}(A'+\varepsilon\hat{H}_i')-G^{-1}\tilde{G}\boldsymbol{p}_2\\\boldsymbol{p}_2.\end{pmatrix} \tag{A.17}$$

in order to obtain

$$H'\boldsymbol{p}^\varepsilon=(G\ \tilde{G})\boldsymbol{p}^\varepsilon=A'+\varepsilon\hat{H}_i'. \tag{A.18}$$

We wish to show that (A.11) holds for $\boldsymbol{p}'=\boldsymbol{p}^\varepsilon$. From (A.16) we observe that $\boldsymbol{p}^\varepsilon-\boldsymbol{p}=\begin{pmatrix}\varepsilon G^{-1}\hat{H}_i'\\0\end{pmatrix}$. So, by Lemma A.2, either $\boldsymbol{p}^\varepsilon=\boldsymbol{p}+\varepsilon\boldsymbol{e}^S$ for some $S\subseteq[n]$ with $i\in S$, or $\boldsymbol{p}^\varepsilon=\boldsymbol{p}-\varepsilon\boldsymbol{e}^S$, where $S\subseteq[n]$ with $i\notin S$. In either case, $\boldsymbol{p}^\varepsilon-\varepsilon\boldsymbol{e}^i\in\{\boldsymbol{p}\pm\varepsilon\boldsymbol{e}^{S'}\}$ for some $S'\subseteq[n]$.

We also immediately observe that $\boldsymbol{p}^\varepsilon$ satisfies (A.14) with equality, and so $\boldsymbol{p}^\varepsilon\in F$. Thus, by definition of F, $\boldsymbol{p}^\varepsilon$ satisfies (A.13). It follows by definition of (A.12) that, for every row in the original systems $H^j\boldsymbol{p}'\geq A^j$ that holds with equality at

$\boldsymbol{p}$, the corresponding shifted equation in (A.11) holds at $\boldsymbol{p}^\varepsilon$. It remains to show that this is also true for rows that are slack at $\boldsymbol{p}$. Let $(H^j)''\boldsymbol{p}' \geq (A^j)''$ be the subsystems of such rows for $j=1, 2$ and observe that, by integrality, it follows that

$$(H^j)''\boldsymbol{p} \geq (A^j)'' + \mathbf{1}. \tag{A.19}$$

Now, applying Observation A.2 to $\boldsymbol{p}^\varepsilon - \boldsymbol{p}$ and (A.19) for $j=1$, we obtain

$$(H^1)''\boldsymbol{p}^\varepsilon = (H^1)''\boldsymbol{p} + (H^1)''(\boldsymbol{p}^\varepsilon - \boldsymbol{p}) \geq (A^1)'' + (1-\varepsilon)\mathbf{1} \geq (A^1)''.$$

And, applying Observation A.2 to $\boldsymbol{p}^\varepsilon - \boldsymbol{p} - \varepsilon\boldsymbol{e}^i$ and (A.19) for $j=2$,

$$(H^2)''(\boldsymbol{p}^\varepsilon - \varepsilon\boldsymbol{e}^i) = (H^2)''\boldsymbol{p} + (H^2)''(\boldsymbol{p}^\varepsilon - \boldsymbol{p} - \varepsilon\boldsymbol{e}^i) \geq (A^2)'' + (1-\varepsilon)\mathbf{1} \geq (A^2)''.$$

So, for every row in the original systems $H^j\boldsymbol{p}' \geq A^j$ that is slack at $\boldsymbol{p}$, the corresponding shifted equation in (A.11) holds for $\boldsymbol{p}^\varepsilon$. This completes the proof that $\boldsymbol{p}^\varepsilon \in P^1 \cap (P^2 + \{\varepsilon\boldsymbol{e}^i\})$. It follows that $\boldsymbol{r} \in D_{u^{1,2[\varepsilon,i]}}(\boldsymbol{p}^\varepsilon)$, which concludes the proof of our first claim.

To see that we can compute $\boldsymbol{p}^\varepsilon$ in polynomial time, recall that the prices at which $\boldsymbol{r}$ is demanded are equivalent to the prices that minimize the Lyapunov function $g(\boldsymbol{p}) = f_{\mathcal{B}'}(\boldsymbol{p}) + \boldsymbol{r}\cdot\boldsymbol{p}$. It is known (see Shioura and Tamura [38, theorem 7.3]) that $f_{\mathcal{B}'}$ (and, thus, g) satisfies the generalized submodularity property

$$f_{\mathcal{B}'}(\boldsymbol{p}) + f_{\mathcal{B}'}(\boldsymbol{q}) \geq f_{\mathcal{B}'}(\boldsymbol{p}\vee\boldsymbol{q}) + f_{\mathcal{B}'}(\boldsymbol{p}\wedge\boldsymbol{q}) \quad (\forall \boldsymbol{p},\boldsymbol{q} \in \mathbb{R}^n),$$

where $\vee$ and $\wedge$ denote the component-wise maximum and minimum, respectively. Hence,

$$h^+(S) := g(\boldsymbol{p} + \varepsilon\boldsymbol{e}^S) - g(\boldsymbol{p}),$$
$$h^-(S) := g(\boldsymbol{p} + \varepsilon\boldsymbol{e}^S) - g(\boldsymbol{p}),$$

are submodular set functions. Hence, in order to determine a price vector $\boldsymbol{p}^\varepsilon \in \{\boldsymbol{p} + \varepsilon\boldsymbol{e}^S \mid S \subseteq [n]\}$ at which $\boldsymbol{r}$ is demanded, we find minimizers S^+ and S^- of h^+ and h^- using SFM and then let

$$\boldsymbol{p}^\varepsilon = \arg\min\{g(\boldsymbol{p} + \varepsilon\boldsymbol{e}^{S^+}), g(\boldsymbol{p} - \varepsilon\boldsymbol{e}^{S^-})\}.$$

Note that we do not require minimal submodular minimizers to find $\boldsymbol{p}^\varepsilon$, and so this step takes $2T(n)$ time, where $T(n)$ is the time it takes to perform submodular minimization on $h^\pm$. We also note that finding $\boldsymbol{p}^\varepsilon$ is analogous to finding a steepest descent direction in Appendix B. □

A.5. Proof of Lemma 7

Here, we prove that applying ShiftProjectUnshift strictly reduces the number of edges in the marginal bids graph $G_{\mathcal{A}}$.

Proof of Lemma 7. First, we show that every edge in $G_{\mathcal{A}'}$ is present in $G_{\mathcal{A}}$. To see this, fix some bid $\boldsymbol{b}$ and note by Observation 1 and Lemma 6 that steps 1–3 do not make $\boldsymbol{b}$ marginal on any new goods.

Second, we prove that, for any multibidder cycle C with cycle-link good i^* and incident label j^*, ShiftProjectUnshift removes at least one of the edges of C from the marginal bids graph. Relabel the goods going around cycle C as $1,\ldots,k$ so that $1 = i^*$ and so that bidder j^* placed the bid marginal between goods 1 and 2. Also, for convenience, index the marginal bid that is marginal between goods i and $i+1$ as "i," and index the marginal bid between goods k and 1 as "k." Thus, $\boldsymbol{b}^1 \in \mathcal{B}^{j^*}$ and $\boldsymbol{b}^k \notin \mathcal{B}^{j^*}$. (In general, the differently labeled bids need not be from different bidders). The existence of a marginal bid between goods i and $i+1$ means that we have an equality

$$b_i^i - p_i = b_{i+1}^i - p_{i+1} \tag{A.20}$$

for $i = 1,\ldots,k-1$, and also

$$b_k^k - p_k = b_1^k - p_1. \tag{A.21}$$

But, if we take the sum of the first $k-1$ equations and cancel, we find

$$\sum_{i=1}^{k-1} b_i^i - \sum_{i=1}^{k-1} b_{i+1}^i = \sum_{i=1}^{k-1}(p_i - p_{i+1}) = p_1 - p_k.$$

So it must hold that

$$\sum_{i=1}^{k-1}(b_i^i - b_{i+1}^i) = b_1^k - b_k^k. \tag{A.22}$$

For any bid $\boldsymbol{b}$ in the bid lists $(\mathcal{B}^j)_{j\in J}$, fixed $j^* \in J$, and $i^* \in [n]$, let

$$\tilde{\boldsymbol{b}} := \begin{cases} \boldsymbol{b} + \frac{1}{10}\boldsymbol{e}^{i^*} & \boldsymbol{b} \in \mathcal{B}^{j^*} \\ \boldsymbol{b} & \text{otherwise.} \end{cases}$$

Suppose we replace each bid $\boldsymbol{b}$ by $\tilde{\boldsymbol{b}}$ as before and recompute the prices. Because prices shift by at most $\frac{1}{10}$, a bid that is not marginal on a pair of goods cannot become marginal on them. If we assume for a contradiction that all edges in C are all still present in G', then we can write down similar expressions to (A.20) and (A.21) (w.r.t. new prices) and then eliminate those prices obtaining a version of (A.22), namely, $\sum_{i=1}^{k-1}(\tilde{b}_i^i - \tilde{b}_{i+1}^i) = \tilde{b}_1^k - \tilde{b}_k^k$. But, by definition of $\tilde{\boldsymbol{b}}$, we know that

$$\sum_{i=1}^{k-1}(\tilde{b}_i^i - \tilde{b}_{i+1}^i) = \sum_{i=1}^{k-1}(b_i^i - b_{i+1}^i) + \frac{1}{10},$$
$$\text{and} \quad \tilde{b}_1^k - \tilde{b}_k^i = b_1^k - b_k^k.$$

and this is inconsistent with (A.22), so we have the required contradiction. □

Appendix B. Finding Market-Clearing Prices

We discuss two iterative steepest descent algorithms, MinUp and LongStepMinUp, from Shioura [37] that determine a minimal minimizer of an $L^\natural$-convex function. Both algorithms use the SFM subroutine described in Section 2 to find the component-wise minimal discrete steepest descent direction. For the second algorithm, we present two methods of computing step lengths and show that both methods yield a polynomial running time in our bidding-language setting.

In order to apply the steepest descent method to our price-finding problem, we define a Lyapunov function g and note in Proposition B.1 that its restriction to $\mathbb{Z}_+^n$ is $L^\natural$-convex. Moreover, Lemma B.1 states that the lowest market-clearing price is integral, and finding it reduces to determining the minimal minimizer of g. This approach generalizes an algorithm used by Ausubel's [1] ascending auction design and Gul and Stacchetti [10] for the task of finding equilibrium prices in single-unit markets.

Let $\mathcal{B}$ be a valid bid list. By Theorem 1, the function $f_{\mathcal{B}}$ defined in (3) is the indirect utility function f_u of some strong-substitutes valuation u. For any strong-substitutes valuation u, the Lyapunov function with regard to indirect utility function f_u and target bundle $\boldsymbol{t}$ is defined as $g_{\boldsymbol{t}}(\boldsymbol{p}) := f_u(\boldsymbol{p}) + \boldsymbol{t}\cdot\boldsymbol{p}$. We suppress the subscript $\boldsymbol{t}$ if it is clear from context. We can use our knowledge of the bids in $\mathcal{B}$ and (3) to express $g_{\boldsymbol{t}}$ for the list $\mathcal{B}$ as

$$g_{\boldsymbol{t}}(\boldsymbol{p}) := f_{\mathcal{B}}(\boldsymbol{p}) + \boldsymbol{t}\cdot\boldsymbol{p} = \sum_{\boldsymbol{b}\in\mathcal{B}} w(\boldsymbol{b}) \max_{i\in[n]_0}(b_i - p_i) + \boldsymbol{t}\cdot\boldsymbol{p}. \tag{B.1}$$

From (B.1), it is clear that we can evaluate g at any price $\boldsymbol{p}$ in time $O(n|\mathcal{B}|)$.

A function $f : \mathbb{Z}^n \to \mathbb{R}$ is $L^\natural$-convex if it satisfies the translation submodularity property

$$f(\boldsymbol{p}) + f(\boldsymbol{q}) \geq f((\boldsymbol{p} - \alpha\mathbf{1}) \vee \boldsymbol{q}) + f(\boldsymbol{p} \wedge (\boldsymbol{q} + \alpha\mathbf{1})) \quad (\forall \boldsymbol{p}, \boldsymbol{q} \in \mathbb{Z}_+^n,\ \forall \alpha \in \mathbb{Z}_+). \tag{B.2}$$

Here, $\vee$ and $\wedge$ denote the component-wise maximum and minimum, respectively.

The following proposition and lemma demonstrate that we can use algorithms for minimizing $L^\natural$-convex functions to find the minimal price $\boldsymbol{p}^*$ at which $\boldsymbol{t}$ is demanded.

Proposition B.1 (Murota et al. [31, 32]). *The Lyapunov function $g_{\boldsymbol{t}}$ restricted to $\mathbb{Z}_+^n$ is $L^\natural$-convex.*

Proof. It is known that an indirect utility function f_u restricted to $\mathbb{Z}^n$ is $L^\natural$-convex if and only if the valuation function u is strong substitutes (cf. Shioura and Tamura [38, theorem 7.1]). Second, it is easy to verify that adding a linear term to an $L^\natural$-convex function preserves $L^\natural$-convexity. As the bid list $\mathcal{B}$ is valid, we have $f_{\mathcal{B}} = f_u$ for some strong-substitutes valuation u, and it follows that the function $f_{\mathcal{B}}(\boldsymbol{p}) + \boldsymbol{t}\cdot\boldsymbol{p}$ is $L^\natural$-convex. □

Lemma B.1. *The Lyapunov function $g_{\boldsymbol{t}}$ with regard to a valid (integral) bid list and any target bundle $\boldsymbol{t}$ is convex, and $\boldsymbol{p}$ is a minimizer of g if and only if it is a market-clearing price for target bundle $\boldsymbol{t}$. Moreover, the minimal minimizer of g is integral.*

Proof. The first statement is immediate from the fact that $f_{\mathcal{B}}$ is convex and $\boldsymbol{t}\cdot\boldsymbol{p}$ is a linear term. To see the second statement, note that, for any market-clearing price $\boldsymbol{p}$ of target bundle $\boldsymbol{t}$, we have $g(\boldsymbol{p}) = u(\boldsymbol{t})$, whereas for any price $\boldsymbol{p}$ at which $\boldsymbol{t}$ is not demanded, we have $g(\boldsymbol{p}) = \max_{\boldsymbol{x}\in D(\boldsymbol{p})}(u(\boldsymbol{p}) - \boldsymbol{x}\cdot\boldsymbol{p}) + \boldsymbol{t}\cdot\boldsymbol{p} > u(\boldsymbol{t})$. Here, u is the valuation function as defined in Section 2, and we use (2). Integrality of the minimal minimizer of g is shown in Murota et al. [32, corollary 4.4]. Alternatively, we can see that this follows from the fact that, if $\boldsymbol{t}$ is demanded at $\boldsymbol{p}$, then $\boldsymbol{t}$ is also demanded at $\lfloor\boldsymbol{p}\rfloor$. To see this, fix a bid $\boldsymbol{b}$ and note that, if $\boldsymbol{b}$ demands good i at $\boldsymbol{p}$, it still demands i at $\lfloor\boldsymbol{p}\rfloor$. Indeed, as $\boldsymbol{b}$ demands i at $\boldsymbol{p}$, we have $b_j - p_j \leq b_i - p_i$ for all goods j, which implies

$$b_j - \lfloor p_j\rfloor < b_j - p_j + 1 \leq b_i - p_i + 1 \leq b_i - \lfloor p_i\rfloor + 1.$$

Because of the integrality of the bids and prices in $\lfloor p\rfloor$, this implies $b_j - \lfloor p_j\rfloor \leq b_i - \lfloor p_i\rfloor$. □

Let $\boldsymbol{p}$ be a point that is dominated by some minimizer of g. Shioura [37] notes that $\boldsymbol{p}$ minimizes g if and only if $g(\boldsymbol{p}) \leq g(\boldsymbol{p} + \boldsymbol{e}^S)$ for every $S \subseteq [n]$. For any integral point $\boldsymbol{p} \in \mathbb{Z}_+^n$ and $S \subseteq [n]$, let the slope function $g'(\boldsymbol{p}; S) := g(\boldsymbol{p} + \boldsymbol{e}^S) - g(\boldsymbol{p})$ denote the amount by which $\boldsymbol{p}$ decreases when moving in the direction of $\boldsymbol{e}^S$. If S minimizes $g'(\boldsymbol{p}; S)$, we call $\boldsymbol{e}^S$ a steepest descent direction. For any integral vector $\boldsymbol{p}$, the $L^\natural$-convexity of g implies that $g'(\boldsymbol{p}; S)$ is an integral submodular function (Shioura and Tamura [38, theorem 7.2]) and hence, there exists a unique component-wise minimal steepest descent direction $\boldsymbol{e}^{S_0}$.

Let $\boldsymbol{p}^*$ be the minimal minimizer of g. If $\boldsymbol{p}$ is dominated by $\boldsymbol{p}^*$ and we move in the minimal steepest descent direction, the point $\boldsymbol{p}+\boldsymbol{e}^{S_0}$ is also dominated by $\boldsymbol{p}^*$. This suggests Algorithm B.1 (called GreedyUpMinimal in Shioura [37]), which iterates a point $\boldsymbol{p}$ by moving by some step $\boldsymbol{e}^{S_0}$, all the while remaining dominated by $\boldsymbol{p}^*$, until $\boldsymbol{p}=\boldsymbol{p}^*$.

Algorithm B.1 (MinUp)
1: Pick a point $\boldsymbol{p} \le \boldsymbol{p}^*$ (e.g., $\boldsymbol{p}=\boldsymbol{0}$).
2: Find the inclusion-wise minimal set $S_0 \subseteq [n]$ minimizing $g'(\boldsymbol{p}; S)$.
3: **if** $S_0 = \emptyset$ **then**
4: **return** $\boldsymbol{p}$.
5: **else**
6: Set $\boldsymbol{p} = \boldsymbol{p} + \boldsymbol{e}^{S_0}$ and go to line 2.

By the existence of the auctioneer's reserve bids, we can initialize $\boldsymbol{p}$ to $\boldsymbol{0}$. For this starting point, running time analysis by Murota and Shioura [29] implies that Algorithm B.1 iterates exactly $\|\boldsymbol{p}^*\|_\infty$ times. As we know that $\boldsymbol{p}^*$ is bounded from above by the component-wise maximum over all bids $\mathcal{B}$, the number of iterations is at most $M := \max_{\boldsymbol{b}\in\mathcal{B}}\|\boldsymbol{b}\|_\infty$. In each iteration of step 2, we can determine S_0 using the SFM subroutine described in Section 2 that finds a minimal minimizer of a given function in time $T(n)$. This leads to the following running time for MinUp.

Theorem B.1 (cf. Murota and Shioura [29]). *The algorithm MinUp finds the component-wise minimal market-clearing price in time $O(MnT(n))$, where $T(n)$ denotes the time it takes to find a minimizer of g'.*

We note that the running times of the two practical SFM algorithms mentioned in Section 2 are given with respect to an upper bound on the absolute value of the objective function. In order to provide such an upper bound for our slope function g', observe that the two points $\boldsymbol{p}$ and $\boldsymbol{p}+\boldsymbol{e}^S$ share a demanded bundle $\boldsymbol{x}$ for any $\boldsymbol{p}$ and $S \subseteq [n]$. As every bid contributes at most one item to $\boldsymbol{t}$ and $\boldsymbol{x}$, this implies $g'(\boldsymbol{p}; S) = (\boldsymbol{t}-\boldsymbol{x})\cdot\boldsymbol{e}^S \le |\mathcal{B}|$.

B.1. Longer Step Sizes

The analysis by Murota and Shioura [29] shows that MinUp performs optimally for an iterative algorithm that is constrained to steps with an L_∞-size of at most 1. As described by Shioura [37], we can, however, exploit monotonicity properties of the function $g'(\boldsymbol{p}; S_0)$ in order to increase step sizes without changing the trajectory of $\boldsymbol{p}$ as the algorithm runs. This reduces the number of SFM subroutine calls, the most expensive part of the algorithm. In particular, if $\boldsymbol{e}^{S_0}$ denotes the minimal steepest descent direction at $\boldsymbol{p}$, we take a single long step $\lambda\boldsymbol{e}^{S_0}$ for some $\lambda \in \mathbb{Z}_+$ that is equivalent to several consecutive steps of MinUp in the same direction $\boldsymbol{e}^{S_0}$.

We follow Shioura [37] in choosing step length

$$\lambda(\boldsymbol{p}, S_0) = \max\{\lambda \in \mathbb{Z}_+ \mid g'(\boldsymbol{p}; S_0) = g'(\boldsymbol{p} + (\lambda-1)\boldsymbol{e}^{S_0}; S_0)\}, \tag{B.3}$$

that is, the farthest distance we can move before the slope

$$g'(\boldsymbol{p} + (\lambda-1)\boldsymbol{e}^{S_0}; S_0) = g(\boldsymbol{p} + \lambda\boldsymbol{e}^{S_0}) - g(\boldsymbol{p} + (\lambda-1)\boldsymbol{e}^{S_0})$$

in the direction of $\boldsymbol{e}^{S_0}$ changes. We give two methods to compute (B.3) in Section B.1.1. This leads to the following algorithm (referred to as GreedyUp-LS in Shioura [37]).

Algorithm B.2 (LongStepMinUp)
1: Pick a point $\boldsymbol{p} \le \boldsymbol{p}^*$ (e.g., $\boldsymbol{p}=\boldsymbol{0}$).
2: Compute inclusion-wise minimal minimizer $S_0 \subseteq [n]$ of $g'(\boldsymbol{p}; S_0)$ using SFM.
3: Determine $\lambda(\boldsymbol{p}, S_0)$ as defined by (B.3), using a method from Section B.1.1.
4: **if** $S_0 \neq \emptyset$ **then**
5: Set $\boldsymbol{p} = \boldsymbol{p} + \lambda(\boldsymbol{p}, S_0)\boldsymbol{e}^{S_0}$ and go to line 2.
6: **return** $\boldsymbol{p}$.

Note that the values of $\boldsymbol{p}$ follow the same trajectory for MinUp and LongStepMinUp. This is an immediate consequence of Lemma B.2, which rests on the monotonicity properties of $g'(\boldsymbol{p}, S_0)$ stated in Proposition B.2.

Proposition B.2 (Shioura [37], theorem 4.16). *Let S_0 and S_0' be minimal steepest descent directions at $\boldsymbol{p}$ and $\boldsymbol{p}+\boldsymbol{e}^{S_0}$, respectively. Then, we have*
1. $g'(\boldsymbol{p}+\boldsymbol{e}^{S_0}; S_0') > g'(\boldsymbol{p}; S_0)$ *or*
2. $g'(\boldsymbol{p}+\boldsymbol{e}^{S_0}; S_0') = g'(\boldsymbol{p}; S_0)$ *and* $S_0 \subseteq S_0'$.

Lemma B.2. *Let S_0 denote the minimal steepest descent at $\boldsymbol{p}$ and let $\lambda(\boldsymbol{p}, S_0)$ be defined by (B.3). Then, $\boldsymbol{e}^{S_0}$ is the minimal steepest descent at $\boldsymbol{p}+(\lambda-1)\boldsymbol{e}^{S_0}$ for any $1 \le \lambda \le \lambda(\boldsymbol{p}, S_0)$.*

Shioura [37] bounds the number of iterations of LongStepMinUp as follows.

Theorem B.2 (Shioura [37], theorem 4.17). *The number of iterations of LongStepMinUp is at most $n\max\{-g'(\boldsymbol{0}; S) \mid S \subseteq [n]\}$.*

Note that $\mathbf{0}$ and $\boldsymbol{e}^{S_0}$ share a demanded bundle $\boldsymbol{x}$, so for any $S \subseteq [n]$, we have

$$g'(\mathbf{0};S) = g(\boldsymbol{e}^S) - g(\mathbf{0}) = f_u(\boldsymbol{e}^S) + \boldsymbol{t}\cdot\boldsymbol{e}^S - f_u(\mathbf{0}) = (\boldsymbol{t}-\boldsymbol{x})\cdot\boldsymbol{e}^S \geq -\sum_{i\in[n]} x_i \geq -|\mathcal{B}|,$$

as each bid contributes at most one unit to the demanded bundle. This implies that LongStepMinUp takes at most $n|\mathcal{B}|$ iterations to find component-wise minimal equilibrium prices.

B.1.1. Computing the Step Length. Fix a price $\boldsymbol{p}$ and let $\boldsymbol{e}^{S_0}$ be the component-wise minimal steepest descent direction at $\boldsymbol{p}$. We describe two methods to compute the step length defined by (B.3). The first method uses binary search and is also suggested in Shioura [37], whereas the second method exploits our knowledge of the bids to determine $\lambda(\boldsymbol{p};S_0)$. Note that we can evaluate $g'(\boldsymbol{p};S_0)$ in time $O(n|\mathcal{B}|)$ as $g'(\boldsymbol{p};S_0) = g(\boldsymbol{p}+\boldsymbol{e}^{S_0}) - g(\boldsymbol{p})$.

Theorem B.3. *The LongStepMinUp algorithm in combination with the binary search and demand change methods has a respective running time of $O(n^2|\mathcal{B}|^2\log M + n|\mathcal{B}|T(n))$ and $O(n^2|\mathcal{B}|^3 + n|\mathcal{B}|T(n))$.*

A description of the two methods as well as a proof of this theorem is provided subsequently. Note that as the two methods have different running time guarantees, the best method in practice is context-specific.

B.1.2. Binary Search. Note that $\lambda(\boldsymbol{p};S_0)$ can be bounded by the total number of unit steps in MinUp, which, in turn, is bounded by M. By Proposition B.2, we have that $g'(\boldsymbol{p};S_0) < g'(\boldsymbol{p}+\lambda\boldsymbol{e}^{S_0};S_0)$ implies $g'(\boldsymbol{p};S_0) < g'(\boldsymbol{p}+\lambda'\boldsymbol{e}^{S_0};S_0)$ for all $\lambda' > \lambda$. Hence, we can apply binary search to find $\lambda(\boldsymbol{p};S_0)$ in time $O(n|\mathcal{B}|\log M)$.

B.1.3. Demand Change. Alternatively, we can exploit our knowledge of the individual bids to determine $\lambda(\boldsymbol{p};S_0)$. We proceed by performing a demand-change procedure, which repeatedly determines the highest value μ for which every bid $\boldsymbol{b}\in\mathcal{B}$ demands the same goods (or a superset thereof) at prices $\boldsymbol{p}+\mu\boldsymbol{e}^{S_0}$ that it demands at $\boldsymbol{p}+\boldsymbol{e}^{S_0}$ and updates $\boldsymbol{p}$ to $\boldsymbol{p}+\mu\boldsymbol{e}^{S_0}$.

Fix a bid $\boldsymbol{b}$ and let I denote the goods it demands at $\boldsymbol{p}$. If $I \not\subseteq S_0$, then $\boldsymbol{b}$ demands the same set of goods $I_{\boldsymbol{b}} = I \setminus S_0$ at all prices $\boldsymbol{p}+\mu'\boldsymbol{e}^{S_0}$ with $\mu' \geq 1$. On the other hand, suppose $I \subseteq S_0$. We define $\mu_{\boldsymbol{b}} := \min_{j\in[n]_0\setminus S_0}((b_i - p_i) - (b_j - p_j))$, where i is any good in I, and consider the set of goods that $\boldsymbol{b}$ demands at prices $\boldsymbol{p}+\mu'\boldsymbol{e}^{S_0}$ with $\mu' \geq 1$. If $1 \leq \mu' < \mu_{\boldsymbol{b}}$, then $\boldsymbol{b}$ demands I; if $\mu' = \mu_{\boldsymbol{b}}$, then $\boldsymbol{b}$ demands a superset of I; and if $\mu' > \mu_{\boldsymbol{b}}$, then $\boldsymbol{b}$ demands none of the goods in I. We define $\mathcal{C}$ to be the list of bids that demand a subset of S_0 at $\boldsymbol{p}$, and let $\mu(\boldsymbol{p},S_0) := \min_{\boldsymbol{b}\in\mathcal{C}}\mu_{\boldsymbol{b}}$. Note that we can determine the value of $\mu(\boldsymbol{p},S_0)$ in time $O(n|\mathcal{B}|)$.

The demand-change procedure takes as input a price $\boldsymbol{p}^0 := \boldsymbol{p}$ and direction $\boldsymbol{e}^{S_0}$ and consists of the following steps. Initially, we set λ to 0. Compute $\mu(\boldsymbol{p},S_0)$ and increment λ by $\mu(\boldsymbol{p},S_0)$. If we have $\lambda = \lambda(\boldsymbol{p}^0,S_0)$, return λ. Otherwise, increment $\boldsymbol{p}$ by $\mu\boldsymbol{e}^{S_0}$ and repeat the procedure with the same value for $\boldsymbol{e}^{S_0}$.

Note that, in order to check whether $\lambda = \lambda(\boldsymbol{p}^0;S_0)$, we can compute $g'(\boldsymbol{p}^0+\mu\boldsymbol{e}^{S_0};S_0)$ and verify that $g'(\boldsymbol{p}^0+\mu\boldsymbol{e}^{S_0};S_0) = g'(\boldsymbol{p}^0;S_0)$, which takes time $O(n|\mathcal{B}|)$. Lemma B.3 proves that the demand-change procedure correctly computes the value of $\lambda(\boldsymbol{p};S_0)$ in time $O(n|\mathcal{B}|^2)$.

Lemma B.3. *The demand-change procedure returns $\lambda(\boldsymbol{p};S_0)$ in at most $|\mathcal{B}|$ iterations.*

Proof. Let K denote the number of iterations in the demand-change procedure and let $\boldsymbol{p}^k, \mu^k$, and C^k be the values of $\boldsymbol{p}, \mu(\boldsymbol{p},S_0)$ and C after the kth iteration. For notational convenience, let $\boldsymbol{p}^0 = \boldsymbol{p}$ and $\mu^0 = 0$. Proposition B.2 implies that it suffices to show

$$g'(\boldsymbol{p};S_0) = g'(\boldsymbol{p}^K - \boldsymbol{e}^S;S_0) \text{ and } g'(\boldsymbol{p};S_0) < g'(\boldsymbol{p}^K;S_0) \tag{B.4}$$

in order to prove the first claim that $\lambda = \lambda(\boldsymbol{p},S_0)$.

First, note that there is a bundle $\boldsymbol{x}$ that is demanded at $\boldsymbol{p}^{k-1}$ and $\boldsymbol{p}^k$ as well as at all prices in between these two points. Indeed, if a bid $\boldsymbol{b}$ demands good i at $\boldsymbol{p}^{k-1}+\boldsymbol{e}^{S_0}$, then $\boldsymbol{b}$ also demands i at the two prices $\boldsymbol{p}^{k-1}$ and $\boldsymbol{p}^k - \boldsymbol{e}^{S_0} = \boldsymbol{p}^{k-1} + (\mu^k - 1)\boldsymbol{e}^{S_0}$ by construction of μ^k. Hence, making use of (2), we get

$$g'(\boldsymbol{p}^{k-1};S_0) = (\boldsymbol{t}-\boldsymbol{x})\cdot\boldsymbol{e}^{S_0} = g'(\boldsymbol{p}^k - \boldsymbol{e}^{S_0};S_0).$$

Second, note that, for every $0 \leq k < K$, we have $g'(\boldsymbol{p}^k - \boldsymbol{e}^{S_0};S_0) = g'(\boldsymbol{p}^k;S_0)$, and for the last iteration K, we have $g'(\boldsymbol{p}^K - \boldsymbol{e}^{S_0};S_0) < g'(\boldsymbol{p}^K;S_0)$. This implies (B.4).

Now, we turn to the second claim, that $K \leq |\mathcal{B}|$. This follows from the fact that $C^{k-1} \supsetneq C^k$ and $|C^0| \leq |\mathcal{B}|$. Indeed, if $\boldsymbol{b} \notin C^{k-1}$, then $\boldsymbol{b}$ demands goods $I \not\subseteq S_0$ at $\boldsymbol{p}^{k-1}$ and $I_{\boldsymbol{b}} = I \setminus S_0$ at any prices $\boldsymbol{p}^{k-1} + \mu\boldsymbol{e}^{S_0}$ for any $\mu \geq 1$, so $b \notin C^k$. Now, suppose $\boldsymbol{b} \in C^{k-1}$ is a bid for which $\mu^{k-1} = \mu_{\boldsymbol{b}}$ (at $\boldsymbol{p}^{k-1}$). Then, we claim that $\boldsymbol{b} \notin C^k$. Indeed, the demanded goods I' of $\boldsymbol{b}$ at $\boldsymbol{p} + (\mu_{\boldsymbol{b}}+1)\boldsymbol{e}^{S_0}$ satisfy $I' \not\subseteq S_0$ by construction of $\mu_{\boldsymbol{b}}$. By the same argument as before, $\boldsymbol{b}$ demands goods not in S_0 at all prices $\boldsymbol{p}+\mu\boldsymbol{e}^{S_0}$ with $\mu \geq \mu_{\boldsymbol{b}} + 1$. □

B.2. Some Practical Improvements

The computation time of MinUp and LongStepMinUp is dominated by the task of finding a minimal set S_0 minimizing $g'(\boldsymbol{p};S)$ using SFM. In practice, we can exploit our direct access to the bids to speed up the computation of S_0 by decreasing the dimensionality of the submodular function to be minimized. The following observations can be seen as a special

case of observations by Ausubel [1]. Fix $\boldsymbol{p} \in \mathbb{Z}^n_+$ and let $\boldsymbol{e}^{S_0}$ be the minimal steepest descent direction at $\boldsymbol{p}$. Note that $\boldsymbol{p}$ and $\boldsymbol{p}+\boldsymbol{e}^{S_0}$ share a demanded bundle $\boldsymbol{x}$ because of the structure of our price space. Hence,

$$g'(\boldsymbol{p}; S_0) = g(\boldsymbol{p}+\boldsymbol{e}^{S_0}) - g(\boldsymbol{p}) = (\boldsymbol{t}-\boldsymbol{x})\cdot \boldsymbol{e}^{S_0},$$

and S_0 minimizes this term if and only if S_0 contains all indices $i \in [n]$ with $t_i < x_i$ and no indices $j \in [n]$ with $t_j > x_j$. In particular, we have $S_0 := \{i \in [n] | t_i < x_i\}$ because of the minimal minimizer property of S_0.

If $\boldsymbol{p}$ has a unique demanded bundle $\boldsymbol{x}$, which is easy to verify, it is straightforward to compute $\boldsymbol{x}$. Hence, in this case, we can determine the minimal steepest descent direction $\boldsymbol{e}^{S_0}$ without performing SFM. In the case that $\boldsymbol{p}$ has at least two demanded bundles, the demanded bundle $\boldsymbol{x}$ shared by $\boldsymbol{p}$ and $\boldsymbol{p}+\boldsymbol{e}^{S_0}$ is unknown. However, if we define index sets I_p and J_p as

$$I_p := \{i \in [n] | x_i > t_i,\ \forall \boldsymbol{x} \in D(\boldsymbol{p})\},$$
$$J_p := \{i \in [n] | x_i \le t_i,\ \forall \boldsymbol{x} \in D(\boldsymbol{p})\},$$

we have $I_p \subseteq S_0$ and $J_p \cap S_0 = \emptyset$. (Note that the inequality in the definition of I_p and J_p is strict and not strict, respectively.) Hence, we can restrict ourselves to minimizing the submodular function $h : [n] \setminus (I_p \cup J_p) \to \mathbb{Z}$ defined by $h(T) = g'(\boldsymbol{p}; T \cup I_p)$ and reduce the dimensionality of the SFM problem from n to $n - |I_p \cup J_p|$. The following lemma shows that computing I_p and J_p is cheap.

Lemma B.4. *I_p and J_p can be computed in time $O(n|\mathcal{B}|)$.*

Proof. Note that $I_p = \{i \in [n] | \min_{\boldsymbol{x} \in D(\boldsymbol{p})} x_i > t_i\}$ and $J_p = \{i \in [n] | \max_{\boldsymbol{x} \in D(\boldsymbol{p})} x_i \le t_i\}$, where we compute the minimum and maximum component-wise. Fix $i \in [n]$. The minimum $\min_{\boldsymbol{x} \in D(\boldsymbol{p})} x_i$ is attained if no marginal bid selects good i. Hence,

$$\min_{\boldsymbol{x} \in D(\boldsymbol{p})} x_i = \sum \{w(\boldsymbol{b}) | \boldsymbol{b} \in \mathcal{B} \text{ is nonmarginal and demands } i\}.$$

Similarly, we claim that $\max_{\boldsymbol{x} \in D(\boldsymbol{p})} x_i$ is attained if good i is selected whenever possible, and thus, $\max_{\boldsymbol{x} \in D(\boldsymbol{p})} x_i$ is the sum of the weights of the bids for which i is demanded. Consider the price perturbation $\boldsymbol{p}'$ defined by $p'_i = p_i$ and $p'_j = p_j + \varepsilon$ for $j \neq i$. Here, $\varepsilon > 0$ is chosen sufficiently small so that every bid that is marginal on i at $\boldsymbol{p}$ is nonmarginal and demands i at $\boldsymbol{p}'$, whereas every nonmarginal bid demanding good i at $\boldsymbol{p}$ is also nonmarginal and demands i at $\boldsymbol{p}'$. This perturbation corresponds to our proposed rule to compute $\max x_i$ by selecting good i whenever possible.

Note that all bundles $\boldsymbol{x}'$ at price $\boldsymbol{p}'$ have the same number of items of i and are also demanded at $\boldsymbol{p}$. As $\boldsymbol{p} \le \boldsymbol{p}'$ and $p_i = p'_i$, the strong-substitutes property implies that there exists $\boldsymbol{x}' \in D(\boldsymbol{p}')$ such that $x_i \le x'_i$ for all $\boldsymbol{x} \in D(\boldsymbol{p})$. In other words, any demanded bundle $\boldsymbol{x}'$ at $\boldsymbol{p}'$ maximizes x_i. □

Appendix C. Hardness of Testing Validity of Unrestricted Bid Lists

Here, we show that the question of whether a list of bids is valid is coNP-complete even when we restrict the problem to unit bids (bids with weights ± 1). We also present a simple algorithm verifying the validity of a given list of positive and negative unit bids that runs in polynomial time if the number of goods or the number of negative bids is bounded by a constant. Let $\mathcal{B}$ denote a list of unit bids and let $\mathcal{B}^+$ and $\mathcal{B}^-$ denote the positive and negative bids.

C.1. Checking Validity Is coNP-Complete

Definition C.1. Given a list of positive and negative bids $\mathcal{B}$, the problem VALID BIDS is to decide whether $\mathcal{B}$ is valid.

Theorem C.1 (Theorem 2). *VALID BIDS is coNP-complete.*

The proof of Theorem C.1 uses an equivalent definition of validity for the list of bids. Define regions in $\mathbb{R}^n_+$ that are generated by a point $\boldsymbol{p}$ and coordinates $i, j \in [n]$ as follows.

$$H^{\boldsymbol{p}}_i := \{\boldsymbol{x} \in \mathbb{R}^n_+ | \boldsymbol{x} \le \boldsymbol{p}, x_i = p_i\}, \tag{C.1}$$
$$F^{\boldsymbol{p}}_{ij} := \{\boldsymbol{x} + \beta \boldsymbol{e}^{[n]} | \beta \in \mathbb{R}_+, \boldsymbol{x} \le \boldsymbol{p}, x_i = p_i \text{ and } x_j = p_j.\}. \tag{C.2}$$

A bid $\boldsymbol{b} = (b_1, \ldots, b_n; b_{n+1})$ is contained in a region $H^{\boldsymbol{p}}_i$ (or $F^{\boldsymbol{p}}_{ij}$) if the "valuation" vector $(b_1, \ldots, b_n)$ consisting of the first n components lies in it. Moreover, we say that $H^{\boldsymbol{p}}_i$ or $F^{\boldsymbol{p}}_{ij}$ is negative for a list of bids if it contains more negative than positive bids; otherwise, it is nonnegative.

Observation C.1. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n_+$, we have $\boldsymbol{x} \in F^{\boldsymbol{y}}_{ij}$ if and only if $x_i - y_i = x_j - y_j$ and $\boldsymbol{x} - (x_i - y_i)\boldsymbol{1} \le \boldsymbol{y}$.

Observation C.2. Fix $i, j \in [n], i \neq j$. Then, containment in the regions given in (C.1) and (C.2) is transitive in the sense that, for any $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n_+$,

- $\boldsymbol{x} \in H^{\boldsymbol{y}}_i$ and $\boldsymbol{y} \in H^{\boldsymbol{z}}_i$ implies $\boldsymbol{x} \in H^{\boldsymbol{z}}_i$.
- $\boldsymbol{x} \in F^{\boldsymbol{y}}_{ij}$ and $\boldsymbol{y} \in F^{\boldsymbol{z}}_{ij}$ implies $\boldsymbol{x} \in F^{\boldsymbol{z}}_{ij}$.

The following definition of valid bids restates the second part of Theorem 1 in terms of regions (C.1) and (C.2).

Definition C.2 (Valid Bids). A list of positive and negative bids is valid if, for any point $\boldsymbol{p} \in \mathbb{R}^n_+$ and two coordinates $i, j \in [n]$, H^p_i and F^p_{ij} are nonnegative.

On the basis of Definition C.2, the proof of Theorem 2 works by reducing the well-known NP-complete problem 3-CNF Satisfiability to Valid Bids by means of an intermediate NP-complete decision problem More Negatives Dominated, which we define in Definition C.3. The NP-completeness of More Negatives Dominated is established in Theorem C.2, and the reduction from More Negatives Dominated to Valid Bids is given as follows as the proof of Theorem 2. For any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we say that $\boldsymbol{x}$ dominates $\boldsymbol{y}$ if $x_i \geq y_i$ for all $i \in \{1, \ldots, n\}$.

Definition C.3. Given two lists P and N of vectors in $\mathbb{R}^n_+$, More Negatives Dominated is the problem of deciding whether there exists a vector $\boldsymbol{z} \in \mathbb{R}^n_+$ that dominates more vectors in N than in P.

Theorem C.2. *The problem More Negatives Dominated is NP-complete.*

Proof. Note that verifying whether a given point $\boldsymbol{z} \in \mathbb{R}^n_+$ dominates more vectors in N than in P can be done in polynomial time. This establishes membership of More Negatives Dominated in the class NP. We show completeness by reducing from 3-CNF Satisfiability. Recall that a Boolean formula ϕ defined on n variables $x_1, \ldots, x_n$ is 3-CNF if

$$\phi = \bigwedge_{i=1}^{m} C_i,$$

where $C_i = L^i_1 \vee L^i_2 \vee L^i_3$ are its clauses and $L^i_j \in \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$ are its literals. Here, $\overline{x_i}$ denotes the negation of x_i. Without loss of generality, we assume that the three variables appearing in each clause are distinct. Given such a 3-CNF formula ϕ, first construct a pair of lists (P_i, N_i) of vectors in $\mathbb{R}^n_+$ for each clause C_i as follows. For notational convenience, $\boldsymbol{e}^{a \cdots z}$ denotes the characteristic vector $\boldsymbol{e}^{\{a, \ldots, z\}}$ of $\{a, \ldots, z\}$.

- If $C_i = x_a \vee x_b \vee x_c$, let $N_i = \{\boldsymbol{e}^{\emptyset}\}$ and $P_i = \{\boldsymbol{e}^{abc}\}$.
- If $C_i = x_a \vee x_b \vee \overline{x_c}$, let $N_i = \{\boldsymbol{e}^{\emptyset}, \boldsymbol{e}^{abc}\}$ and $P_i = \{\boldsymbol{e}^{ab}\}$.
- If $C_i = x_a \vee \overline{x_b} \vee \overline{x_c}$, let $N_i = \{\boldsymbol{e}^{\emptyset}, \boldsymbol{e}^{ab}, \boldsymbol{e}^{ac}\}$ and $P_i = \{\boldsymbol{e}^{a}, \boldsymbol{e}^{abc}\}$.
- If $C_i = \overline{x_a} \vee \overline{x_b} \vee \overline{x_c}$, let $N_i = \{\boldsymbol{e}^{a}, \boldsymbol{e}^{b}, \boldsymbol{e}^{c}, \boldsymbol{e}^{abc}\}$ and $P_i = \{\boldsymbol{e}^{ab}, \boldsymbol{e}^{ac}, \boldsymbol{e}^{bc}\}$.

Finally, let N be the sum of the N_i and let P be the sum of the P_i together with $m-1$ copies of $\boldsymbol{e}^{\emptyset}$. This completes our reduction from 3-CNF Satisfiability to More Negatives Dominated. Clearly, P and N can be constructed from ϕ in polynomial time. To show correctness of the reduction, associate with every vector $\boldsymbol{z} \in \mathbb{R}^n_+$ a truth assignment β_z (on Boolean variables $x_1, \ldots, x_n$) that sets x_i to true if $z_i < 1$ and false otherwise.

Observation C.3. Let $\boldsymbol{z} \in \mathbb{R}^n_+$ be a point and β_z be its associated truth assignment. For any $i \in \{1, \ldots, m\}$, $\boldsymbol{z}$ dominates one more point in N_i than in P_i if β_z satisfies C_i and an equal number of points in N_i and P_i if β_z does not satisfy C_i.

Suppose ϕ is a satisfiable 3-CNF formula with satisfying truth assignment β. Then, define $\boldsymbol{z} \in \mathbb{R}^n_+$ by

$$z_i = \begin{cases} 0 & \text{if } \beta[x_i] = \text{true}, \\ 1 & \text{else.} \end{cases}$$

Hence, β is the truth assignment associated with z. By Observation C.3, $\boldsymbol{z}$ dominates one more point in N_i than in P_i for each i, so because of the additional $m-1$ origin points $\boldsymbol{e}^{\emptyset}$ added to P, $\boldsymbol{z}$ dominates exactly one more point in N than in P. Conversely, suppose there exists a point $\boldsymbol{z} \in \mathbb{R}^n_+$ that dominates more points in N than in P, and let β_z be its corresponding truth assignment. Then, $\boldsymbol{z}$ dominates at least m more points in N than in $\sum_{i=1}^m P_i$. By Observation C.3, we know that $\boldsymbol{z}$ dominates at most one more point in N_i than in P_i, which implies that this is the case for each i. Hence, by the same observation, β_z satisfies all clauses of ϕ and, thus, ϕ itself.

Proof of Theorem 2. In order to show that a list of bids $\mathcal{B}$ is not valid, it suffices to provide a certificate in the form of a point $\boldsymbol{p}$ and coordinates $i, j \in [n]$. We can verify in polynomial time whether H^p_i or F^p_{ij} is negative. This establishes membership of Valid Bids in coNP.

Let (N, P) be an instance of More Negatives Dominated and n be the dimension of its vectors. We construct an $(n+1)$-dimensional instance $\mathcal{B}$ of Valid Bids such that $(N, P) \in$ More Negatives Dominated if and only if $\mathcal{B} \notin$ Valid Bids as follows. Let the negative and positive bids of $\mathcal{B}$ be given by prepending a component to the vectors as follows:

$$\mathcal{B}^- := \{(1, \boldsymbol{v}) | \boldsymbol{v} \in N\}$$
$$\text{and } \mathcal{B}^+ := \{(1, \boldsymbol{w}) | \boldsymbol{w} \in P\} + \{(0, \boldsymbol{v}) | \boldsymbol{v} \in N\} + \{(1, \boldsymbol{v}) + \boldsymbol{e}^{[n+1]} | \boldsymbol{v} \in N\}.$$

Clearly, $\mathcal{B} = \mathcal{B}^+ + \mathcal{B}^-$ can be constructed efficiently. Note that all F^x_{ij} and all H^x_i with $i \neq 0$ are nonnegative by construction. Indeed, fix F^x_{ij} and suppose it contains r negative bids. Then, for each such bid $(1, \boldsymbol{v})$, where $\boldsymbol{v} \in N$, there exists a positive bid $(1, \boldsymbol{v}) + \boldsymbol{e}^{[n+1]}$ that is contained in F^x_{ij} so that the region is nonnegative. Analogously, for any H^x_i with $i \neq 0$, each negative bid $(1, \boldsymbol{v})$ has a corresponding positive bid $(0, \boldsymbol{v})$. Further, as H^x_0 contain no negative points unless $x_0 = 1$, the correctness of our reduction, thus, follows from the fact that $\boldsymbol{z}$ dominates r points from N and s points from P if and only if H^x_0 with $\boldsymbol{x} = (1, \boldsymbol{z})$ contains r negative and s positive bids. □

C.2. A Simple Algorithm for Testing Validity

In Section C.1, we show that checking the validity of a bid list in the general case is coNP-complete. Hence, we cannot expect an efficient algorithm for the task of checking validity of a list of bids in the general case. Here, we present a simple algorithm that tests validity of a given list of bids in polynomial time if the number of goods or the number of negative bids is bounded by a constant. Such constraints may be reasonable in certain economic settings.

C.2.1. The Algorithm. Our procedure rests on Theorem C.3, which reduces the validity condition from Definition C.2 to a finite number of checks. Lemma C.1 reduces the number of checks further. Together, the two results immediately yield Algorithm C.1, whose running time is given in Theorem C.4. For any list of bids U, define the minimal dominating vector $md(U)$ of U as the component-wise maximum over the valuation vectors of the bids in U so that $md(U)_i = \max_{\boldsymbol{b}\in U} b_i$ for all $i \in [n]$. We note that, if all bids of U agree on some coordinate i (that is, if $b_i = b'_i$ for all $\boldsymbol{b}, \boldsymbol{b}' \in U$), they lie in H_i^x for large enough $\boldsymbol{x}$, and $md(U)$ is the minimal such $\boldsymbol{x}$ so that H_i^x contains all points in U. Similarly, for any list of bids U and $i \in [n]$, define $mdF(i, U)$ as $mdF(i,U) := \min_{\boldsymbol{b}\in U} b_i\mathbf{1} + md(\{\boldsymbol{b} - b_i\mathbf{1} | \boldsymbol{b} \in U\})$. We note that, if, for some $i \neq j$, the set U satisfies $b_i - b_j = b'_i - b'_j$ for all $\boldsymbol{b}, \boldsymbol{b}' \in U$, then the bids in U lie in F_{ij}^x for small enough $\boldsymbol{x}$, and it follows from Observation C.1 that $mdF(i,U) = mdF(j,U)$ is the maximal such $\boldsymbol{x}$.

Theorem C.3. *A list of bids $\mathcal{B}$ is valid if and only if the following two conditions hold:*

1. *For every set $U \subseteq \mathcal{B}^-$ of negative bids that agree on the ith coordinate, that is, $b_i = b'_i \ \forall b, b' \in U$, the region $H_i^{md(U)}$ is nonnegative.*
2. *For every set $U \subseteq \mathcal{B}^-$ of negative bids that satisfies $b_i - b'_i = b_j - b'_j$ for all $b, b' \in U$ and some $i \neq j$, the region $F_{ij}^{mdF(i,U)}$ is nonnegative.*

Proof. The implication is immediate by Definition C.2. Conversely, suppose $\mathcal{B}$ satisfies the second condition. First, we show that condition 1 of Definition C.2 is satisfied. Fix H_i^x and let $U = \mathcal{B}^- \cap H_i^x$ be the set of negative bids in H_i^x and $\boldsymbol{y} = md(U)$. Then, by construction, we have $\boldsymbol{b} \in H_i^y$ for all $\boldsymbol{b} \in U$ and $\boldsymbol{y} \in H_i^x$. As H_i^y has $|U|$ negative bids, it also contains at least $|U|$ positive bids by assumption, and by Observation C.2, these positive bids are also in H_i^x. Condition 2 is shown analogously. Fix F_{ij}^x and let U be the negative bids in this region and $\boldsymbol{z} = mdF(U)$. Suppose $\boldsymbol{b} \in F_{ij}^z$ for every $\boldsymbol{b} \in U$ and $\boldsymbol{z} \in F_{ij}^x$. Then, Observation C.2 implies that F_{ij}^x is nonnegative. To see that $\boldsymbol{b} \in F_{ij}^z$ for every $\boldsymbol{b} \in U$, we verify the conditions of Observation C.1. First,

$$z_i - z_j = \min_{\boldsymbol{b}\in U} b_i - \left(\min_{\boldsymbol{b}\in U} b_i + \max_{\boldsymbol{b}\in U}(b_j - b_i)\right) = \max_{\boldsymbol{b}\in U}(b_j - b_i) = b_j - b_i$$

for all $\boldsymbol{b} \in U$ as the difference $b_j - b_i$ is the same for all $\boldsymbol{b} \in F_{ij}^z$. Second, for any $k \in [n]$ and $\boldsymbol{b} \in U$, we have

$$(\boldsymbol{b} - (b_i - z_i)\mathbf{1})_k = z_i + b_k - b_i \leq z_i + \max_{\boldsymbol{b}\in U}(b_k - b_i) = z_k.$$

We can verify $\boldsymbol{z} \in F_{ij}^x$ similarly. □

Lemma C.1. *For any list of negative bids $U \subseteq \mathcal{B}^-$, there exists a list $U' \subseteq U$ with $|U'| \leq n$ so that $md(U') = md(U)$. Moreover, for each coordinate $i \in [n]$, there exists a list $U'' \subseteq U$ with $|U''| \leq n+1$ so that $mdF(i,U'') = mdF(i,U)$.*

Proof. Let U' be a list of bids $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^n$, where $\boldsymbol{u}^k$ is a bid from U that maximizes the kth component, that is, $\boldsymbol{u}^k \in \arg\max_{\boldsymbol{v}\in U} v_k$. Then, $md(U') = md(U)$ by construction. Second, fix $i \in [n]$ and let U'' be a list of bids $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^{n+1}$, where $\boldsymbol{u}^1 \in \arg\min_{\boldsymbol{b}\in U} b_i$ and $\boldsymbol{u}^k \in \arg\max_{\boldsymbol{b}\in U}(b_k - b_i)$ for $k \in \{2, \ldots, n+1\}$. Then, $mdF(U'') = mdF(U)$ by construction, and we are done. □

Algorithm C.1 (Checking the Validity of a Bid List $\mathcal{B}$)

1: **for all** subsets $U \subseteq \mathcal{B}^-$ of negative bids with $|U| \leq n+1$ **do**
2: **for all** $i, j \in [n]$ **do**
3: Verify the two conditions given in Theorem C.3.

Theorem C.4. *Algorithm C.1 runs in time $O\left(\binom{|\mathcal{B}^-|}{n} n^3 |\mathcal{B}|\right)$. Moreover, if $|\mathcal{B}^-| \leq k$ or $n \leq k$, the algorithm's running time is polynomial in the input size $n|\mathcal{B}|$.*

Proof. There are $\binom{|\mathcal{B}^-|}{n+1}$ subsets U of $\mathcal{B}^-$ for the algorithm to iterate over, whereas there are n^2 possibilities for the index pair i, j. For each U and i, j, checking the conditions in Theorem C.3 takes $n|\mathcal{B}|$ time. This implies the running time. Suppose the number of negative bids $|\mathcal{B}^-| \leq k$ is bounded by some $k \in \mathbb{N}$. Then, $\binom{|\mathcal{B}^-|}{n+1}$ is constant and the running time of our algorithm is $O(n^3(|V| + |W|))$. Second, if the number of goods $n \leq k$ is bounded, we have $\binom{|\mathcal{B}^-|}{n+1} \leq |\mathcal{B}^-|^{k+1}$ and, hence, a running time of $O(|\mathcal{B}^-|^{k+1}|\mathcal{B}|)$. □

Endnotes

[1] In the Bank of England's auction, the bidders are commercial banks, etc.; each good is a loan secured against one of n different specified qualities of collateral (so the prices are interest rates); and the quantity is the amount of the loan (in £).

[2] Specifically, the auction finds the lowest such price vector. Because the bids automatically express strong-substitutes preferences for all bidders (see Baldwin and Klemperer [3]), there exist equilibrium price vectors and also a unique one among them at which every good's price is lowest.

[3] See Klemperer [17, appendix 1E] for how to convert supply functions into bid lists. (The bids into which the auctioneer's supply functions would be converted would ensure that it sells more units on a good when prices are high, analogous to a buyer buying fewer units in this case.) Although the auctioneer's preferences could equivalently be represented as a list of bids of the kind made by the bidders, describing them simply as two-dimensional graphs of "supply schedules" was an important feature of the auction design: participants' ability to express their preferences in the ways that are most natural for them is crucial to getting an auction accepted for practical use, to getting bidders to participate, and to the auction working efficiently. The Bank of England's original program restricted to $n = 2$. Since 2014, its program permits much larger n (it is currently being run with $n = 3$), and it also allows richer forms of preferences to be expressed by the auctioneer (but not by the bidders, whose preference expression has, by contrast, been restricted in recent auctions). A variant that allowed for bidders' budget constraints (hence, nonquasilinear preferences) was programmed for the Government of Iceland in 2015–2016. For additional information about the strong-substitutes product-mix auction and other variants of the product-mix auction that allow for divisible goods and budget constraints, etc., we refer to http://pma.nuff.ox.ac.uk, Baldwin and Klemperer [3], and Klemperer [15–17].

[4] Strong-substitutes preferences are those that would be still be gross substitutes preferences if we treated every unit of every good as a separate good. Such preferences have many attractive properties; they mean, for example, that, if the price of any one good increases and the demand for it decreases, then the demand for all other goods can increase by at most the amount of that decrease. Strong substitutability is equivalent to $M^\natural$-concavity (Shioura and Tamura [38]) in our setting with valuations defined on nonnegative and bounded domains.

[5] The ability to express such preferences was not thought necessary in the Bank of England's application but might be useful in closely related environments (see Klemperer [17]).

[6] Klemperer [16] stated this result for the case of multiple units of each of two goods. See Baldwin and Klemperer [3] and Lin and Tran [19] for working papers proving (different) generalizations of this result. (A prior proof of the result itself was presented at a workshop as part of Baldwin et al. [4].)

[7] This exploits a result that market-clearing prices minimize the difference between two linear programs (one for the positive bids and one for the negative bids).

[8] We have $\alpha(n) \leq 4$ for any $n \leq 2^{2^{2^{16}}}$.

[9] A function is concave-extensible if its concave closure agrees with the function at every integral point. See Shioura and Tamura [38] for details and how this assumption is natural for valuation functions.

[10] Formally, we can associate every multiunit valuation function u on a bounded effective domain with a single-unit valuation function $\hat{u} : \{0,1\}^N \to \mathbb{R}$ as follows. Let U be an upper bound on the entries of the bundles in the domain of u. We define the domain $\hat{u}$ as $N = \{(i,\beta) | i \in [n], \beta \in [U]\}$ and let $\hat{u}(\hat{\boldsymbol{x}}) = u(\boldsymbol{x})$, where $\boldsymbol{x}$ is given by $x_i = \sum_{\beta=1}^{U} \hat{x}_{i,\beta}$. Valuation u is strong substitutes if $\hat{u}$ is gross substitutes. See Shioura and Tamura [38] for details.

[11] Equivalent conditions for validity are provided in concurrent work (Baldwin and Klemperer [3, proposition 4.14] in a more general setting. There, also, it is shown that condition (2) is equivalent to the existence of a strong-substitutes valuation u satisfying $D_{\mathcal{B}} = D_u$ (that is, the second part of condition (3)). Baldwin and Klemperer [3] do not work with the indirect utility function, but do additionally show that validity is equivalent to $D_{\mathcal{B}}$ satisfying the "law of demand": an increase in the price of only one good between prices at which demand is unique leads to a weak decrease in demand for the good whose price has changed with equality in this demand if and only if the full demand vector is unchanged. Baldwin et al. [5] additionally show that, when $\mathcal{B}$ is valid, the demand $D_{\mathcal{B}}$ can be given as the Minkowski difference between the demand associated with the set of positive bids in $\mathcal{B}$ and the demand associated with the set of negative bids in $\mathcal{B}$. (The work of Lin and Tran [19], mentioned in Endnote 6, does not consider characterizations of validity as, in their framework, invalid combinations do not arise.)

[12] Of course, any positive reserve price can be reflected by appropriately locating the auctioneer's bids.

[13] See Murota [25, theorem 6.13(4)].

[14] Note that $u^{1,2}$ can be expressed as $u^{1,2}(\boldsymbol{x}) = \max_{\boldsymbol{y} \in \mathbb{Z}^n} (u^1(\boldsymbol{y}) + u^2(\boldsymbol{y} - \boldsymbol{x}))$ for all $\boldsymbol{x} \in \mathbb{Z}^n$. This is stated, for instance, in Baldwin and Klemperer [2, footnote 2] and is related to the aggregate cost function in Murota [25, chapter 11]. We do not make use of this way of expressing $u^{1,2}$ in the following proofs.

References

[1] Ausubel LM (2006) An efficient dynamic auction for heterogeneous commodities. *Amer. Econom. Rev.* 96(3):602–629.
[2] Baldwin E, Klemperer P (2019) Understanding preferences: "Demand types," and the existence of equilibrium with indivisibilities. *Econometrica* 87(3):867–932.
[3] Baldwin E, Klemperer P (2021) Proof that the product-mix auction bidding language can represent any substitutes preferences. Working paper. Accessed December 31, 2021, https://www.nuffield.ox.ac.uk/economics/Papers/2021/2021-W05_subsproof.pdf.
[4] Baldwin E, Goldberg PW, Klemperer P (2016) Tropical intersections and equilibrium (day 2 slides). Hausdorff School on Tropical Geometry and Economics. Accessed July 1, 2019, http://people.math.gatech.edu/~jyu67/HCM/Baldwin2.pdf.
[5] Baldwin E, Bichler M, Fichtl M, Klemperer P (2022) Strong substitutes: Structural properties, and a new algorithm for competitive equilibrium prices. *Math. Programming* 1–33.
[6] Chakrabarty D, Jain P, Kothari P (2014) Provable submodular minimization using Wolfe's algorithm. Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, eds. *Adv. Neural Inform. Processing Systems*, vol. 802–809 (Curran Associates, Inc., Cambridge, MA).
[7] Chakrabarty D, Lee YT, Sidford A, Wong SCW (2017) Subquadratic submodular function minimization. *Proc. 49th Annual ACM SIGACT Sympos. Theory Comput.* (ACM, New York), 1220–1231.
[8] Danilov V, Koshevoy G, Murota K (2001) Discrete convexity and equilibria in economies with indivisible goods and money. *Math. Social Sci.* 41(3):251–273.
[9] Goldberg PW, Lock E, Marmolejo-Cossío F (2022) Learning strong substitutes demand via queries. *ACM Trans. Econom. Comput.* 10(2):1–22.
[10] Gul F, Stacchetti E (2000) The English auction with differentiated commodities. *J. Econom. Theory* 92(1):66–95.
[11] Iwata S (2008) Submodular function minimization. *Math. Programming* 112(1):45–64.

[12] Jiang H (2021) Minimizing convex functions with integral minimizers. *Proc. 2021 ACM-SIAM Sympos. Discrete Algorithms* (SIAM), 976–985.
[13] Joswig M (2021) *Essentials of Tropical Combinatorics*, Graduate Studies in Mathematics, vol. 219 (American Mathematical Society, Providence, RI).
[14] Kelso AS, Crawford VP (1982) Job matching, coalition formation, and gross substitutes. *Econometrica* 50(6):1483–1504.
[15] Klemperer P (2008) A new auction for substitutes: Central bank liquidity auctions, the U.S. TARP, and variable product-mix auctions. Working paper, Nuffield College, Oxford, UK.
[16] Klemperer P (2010) The product-mix auction: A new auction design for differentiated goods. *J. Eur. Econom. Assoc.* 8(2–3):526–536.
[17] Klemperer P (2018) Product-mix auctions. Working Paper 2018-W07, Nuffield College, Oxford, UK.
[18] Lee YT, Sidford A, Wong SCW (2015) A faster cutting plane method and its implications for combinatorial and convex optimization. *Proc. 2015 IEEE 56th Annual Sympos. Foundations Comput. Sci.* (IEEE Computer Society, Washington, DC), 1049–1065.
[19] Lin B, Tran NM (2017) Linear and rational factorization of tropical polynomials. Preprint, submitted July 11, https://arxiv.org/abs/1707.03332.
[20] Maclagan D, Sturmfels B (2015) *Introduction to Tropical Geometry*, Graduate Studies in Mathematics, vol. 161 (American Mathematical Society, Providence, RI).
[21] Mas-Colell A, Whinston MD, Green JR (1995) *Microeconomic Theory*, vol. 1 (Oxford University Press, New York).
[22] McCormick ST (2005) Submodular function minimization. Aardal K, Nemhauser G, Weismantel R, eds. *Discrete Optimization*, *Handbooks in Operations Research and Management Science*, vol. 12 (Elsevier, Amsterdam), 321–391.
[23] Mikhalkin G (2004) Decomposition into pairs-of-pants for complex algebraic hypersurfaces. *Topology* 43(5):1035–1065.
[24] Milgrom P, Strulovici B (2009) Substitute goods, auctions, and equilibrium. *J. Econom. Theory* 144(1):212–247.
[25] Murota K (2003) *Discrete Convex Analysis*, SIAM Monographs on Discrete Mathematics and Applications.
[26] Murota K (2003) On steepest descent algorithms for discrete convex functions. *SIAM J. Optim.* 14(3):699–707.
[27] Murota K (2016) Discrete convex analysis: A tool for economics and game theory. *J. Mechanism Inst. Design* 1(1):151–273.
[28] Murota K, Shioura A (1999) M-convex function on generalized polymatroid. *Math. Oper. Res.* 24(1):95–105.
[29] Murota K, Shioura A (2014) Exact bounds for steepest descent algorithms of L-convex function minimization. *Oper. Res. Lett.* 42(5):361–366.
[30] Murota K, Tamura A (2001) Application of M-convex submodular flow problem to mathematical economics. *Proc. 12th Internat. Sympos. Algorithms Comput.* (Springer-Verlag, Berlin, Heidelberg), 14–25.
[31] Murota K, Shioura A, Yang Z (2013) Computing a Walrasian equilibrium in iterative auctions with multiple differentiated items. Cai L, Cheng SW, Lam TW, eds. *Algorithms and Computation* (Springer, Berlin, Heidelberg), 468–478.
[32] Murota K, Shioura A, Yang Z (2016) Time bounds for iterative auctions: A unified approach by discrete convex analysis. *Discrete Optim.* 19:36–62.
[33] Paes Leme R (2017) Gross substitutability: An algorithmic survey. *Games Econom. Behav.* 106:294–316.
[34] Paes Leme R, Wong SCW (2017) Computing Walrasian equilibria: Fast algorithms and structural properties. *Proc. 28th Annual ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 632–651.
[35] Schrijver A (1998) Theory of Linear and Integer Programming, *Interscience Series in Discrete Mathematics and Optimisation* (John Wiley & Sons).
[36] Shioura A (1998) Minimization of an M-convex function. *Discrete Appl. Math.* 84(1):215–220.
[37] Shioura A (2017) Algorithms for L-convex function minimization: Connection between discrete convex analysis and other research fields. *J. Oper. Res. Soc. Japan* 60(3):216–243.
[38] Shioura A, Tamura A (2015) Gross substitutes condition and discrete concavity for multi-unit valuations: A survey. *J. Oper. Res. Soc. Japan* 58(1):61–103.
[39] Tarjan RE, van Leeuwen J (1984) Worst-case analysis of set union algorithms. *J. ACM* 31(2):245–281.
[40] van der Walt S, Colbert SC, Varoquaux G (2011) The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Engrg.* 13(2):22–30.