Noname manuscript No.
(will be inserted by the editor)

Strong Substitutes: Structural Properties, and a New Algorithm for Competitive Equilibrium Prices

Elizabeth Baldwin · Martin Bichler ·
Maximilian Fichtl · Paul Klemperer

the date of receipt and acceptance should be inserted later

October 2021

Received: date / Accepted: date

Abstract We show the Strong Substitutes Product-Mix Auction (SSPMA) bidding language provides an intuitive and geometric interpretation of strong substitutes as Minkowski differences between sets that are easy to identify. We prove that competitive equilibrium prices for agents with strong substitutes preferences can be computed by minimizing the difference between two linear programs for the positive and the negative bids with suitably relaxed resource constraints. This also leads to a new algorithm for computing competitive equilibrium prices which is competitive with standard steepest descent algorithms in extensive experiments.

Keywords Competitive equilibrium · Walrasian equilibrium · Strong substitutes · Product-Mix auction · Envy-free prices · Indivisible goods · Equilibrium computation · DC programming · Auction theory · Algorithms · product mix auction

Corresponding author:
Maximilian Fichtl. Email: max.fichtl@tum.de, phone: +4917699811410.

Elizabeth Baldwin
Dept. of Economics and Hertford college, Oxford University

Martin Bichler
Dept. of Computer Science, Technical University of Munich

Maximilian Fichtl
Dept. of Computer Science, Technical University of Munich

Paul Klemperer
Dept. of Economics and Nuffield College, Oxford University

1 Introduction

This paper shows that for an important and widely-studied class of problems–those for which agents have strong substitutes valuations over multiple units of multiple differentiated goods–competitive-equilibrium prices can be found by considering two linear programs. Specifically, we relax resource constraints on both programs in the same way, and find the relaxation that minimizes the difference between the objectives of the two programs; the dual prices of one of these relaxed programs are competitive equilibrium prices. We derive this result by using the geometric representation of preferences provided by the Strong Substitutes Product-Mix Auction (SSPMA) bidding language. This then allows us to develop an efficient algorithm to find the competitive equilibrium prices when preferences are represented this way. Since, as we detail below, the SSPMA language is a natural way for agents to express their preferences, our algorithm is a practical way to find competitive equilibrium prices for strong substitutes.

Our paper also provides a novel algorithm to find the prices in an SSPMA, since these are prices that would be competitive equilibrium prices for the given aggregate supply if bidders had bid their actual values.[1] Participants in SSPMAs make bids that express "strong-substitutes" preferences for multiple units of multiple, differentiated, indivisible goods. Strong-substitutes preferences are those that would be ordinary substitutes preferences if every unit of every good were treated as a separate good (Milgrom and Strulovici, 2009). They are an extension of gross substitutes preferences (Kelso and Crawford, 1982) from single-unit to multi-unit, multi-item markets, and are equivalent to $M^\natural$-concavity (Danilov et al., 2001; Murota, 2016; Shioura and Tamura, 2015). They have many attractive properties. In particular, all agents having strong-substitutes preferences is a sufficient condition for the existence of competitive equilibrium prices in markets with indivisible goods.

Furthermore, even though strong substitutes are a small subset of the set of all possible valuation functions of a bidder, they are practically relevant for various applications such as auctions used by the Bank of England (Klemperer, 2008, 2018). So a significant amount of theoretical literature has been devoted to markets where participants have these valuations (Ausubel, 2006; Baldwin and Klemperer, 2019; Murota and Tamura, 2003; Paes Leme, 2017).

Importantly, valid bids in the SSPMA bidding language permits the specification of *precisely* the set of preferences that are strong substitutes, and indeed is the *only* language that is known to do this.[2] As we will see, it is

[1] Product-Mix Auctions give envy-free allocations to bidders who express their valuations truthfully. The auctioneer can express its own preferences, and if all the bidders and the auctioneer express their true valuations (the Bank of England does in its role as a product-mix auctioneer, and bidders approximate this if no one bidder is too large) then the auction yields a competitive equilibrium.

[2] See Baldwin and Klemperer (2016, 2021). By contrast, Ostrovsky and Paes Leme (2015) show Hatfield and Milgrom (2005)'s endowed assignment messages cannot express all strong substitute valuations, Fichtl (2021) likewise shows Milgrom (2009)'s (integer) assignment messages cannot express all strong substitute valuations, and Tran (2019) shows that it is

also parsimonious, or “compact”, in that many valuations can be expressed using only a small number of simple bids.[3] Finally, it expresses valuations in a natural way, which can be understood and analyzed geometrically; we show aggregate demand is the Minkowski difference between two easily identified demand sets.

1.1 The Strong Substitutes Product-Mix Auction (SSPMA)

There is significant literature on computing competitive equilibria with strong substitutes valuations. See, for example, Kelso and Crawford (1982), Bikhchandani and Mamer (1997), Gul and Stacchetti (1999), Murota and Tamura (2003), Ausubel (2006), Nisan and Segal (2006), Milgrom and Strulovici (2009), Paes Leme (2017), Bichler et al. (2020), and Paes Leme and Wong (2020).[4] The interest in strong substitutes is due to the fact that it captures practically relevant valuations for indivisible goods, but the allocation problem can be solved in polynomial time and Walrasian competitive equilibrium prices always exist, which is not the case for general valuations (Bikhchandani and Ostroy, 2002).

Prior literature requires either value oracles for exponentially many bundles, or demand oracles. Demand oracles can be understood as indirect or iterative mechanisms, where bidders reveal their demand correspondence for a set of prices specified by the auctioneer. So in a large market with many goods that is organized as a sealed-bid auction, the auctioneer needs to perform an exponential number of value queries for each bidder before the allocation algorithm can be run. Such enumerative (XOR) bid languages are used in spectrum auctions, but can lead to “missing bids” problems, which can significantly affect prices, and also create efficiency losses (Bichler et al., 2013).[5]

The SSPMA was developed by Klemperer (2008) for the Bank of England to provide liquidity to financial institutions by auctioning loans to them. The SSPMA is neither based on a value nor a demand oracle.[6] A collection of bids specifies a large number of package values, which mitigates the missing bids problem. This type of preference elicitation permits efficient ways to compute Walrasian prices, and allows us to uncover new properties of strong substitutes valuations.

not possible to express all strong substitute valuations as combinations of weighted ranks of matroids on a ground set bounded by the number of goods.

[3] See Goetzendorff et al. (2015) for a discussion of compactness.

[4] Paes Leme and Wong (2020) provides the fastest algorithm for value oracles and a new algorithm for aggregate demand queries. However, the latter is different in spirit to our paper which addresses a market design for applications such as the Bank of England’s.

[5] Bidders who do not submit the very large number of bids required to fully specify their valuations are treated as if they place no value on the packages they fail to bid for.

[6] If a demand oracle is what is available, a conversion to SSPMA is available via Goldberg et al. (2020)’s algorithm which computes the (unique) list of bids corresponding to a bidder’s demand preferences, given access to either a demand or a valuation oracle.

Each bidder makes a set of bids, each of which is a vector $\mathbf{b}$, incorporating an integer weight $w(\mathbf{b})$. Each bidder's set of bids is interpreted as specifying a quasi-linear utility function over multiple units of each of n goods plus money. A bid in which $w(\mathbf{b}) > 0$ (a "positive" bid) is simply interpreted as a bid for up to, but not more than, $w(\mathbf{b})$ units, in total, of the goods $i = 1, \ldots, n$, in which the expressed value of receiving x_i units of good i is $x_i \cdot b_i$.

Example 1. A bidder might be interested in 2 units, and be willing to pay up to price 2 for each unit of good 1, but only up to price 1 for each unit of good 2. These preferences can be implemented by a single bid $\mathbf{b} = (2, 1)$ with $w(\mathbf{b}) = 2$. Figure 1 shows how the bid $\mathbf{b}$ divides price space into three regions: for example, if the price vector $\mathbf{p} = (p_1, p_2)$ lies in the region labeled as "(2, 0) demanded" then, at this price vector, receiving the bundle $(2, 0)$ maximizes the bidder's utility among all feasible bundles. The black lines mark the borders at which the demanded bundle changes. If prices lie exactly on the boundary of two or more regions, then the set of demanded bundles is given by the discrete convex hull of the bundles demanded in the adjacent regions. For example, if $\mathbf{p} = (2, 4)$, the bidder demands bundles $(0, 0)$, $(1, 0)$ or $(2, 0)$.

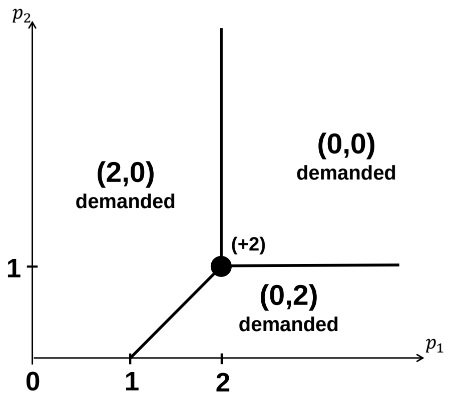

Fig. 1 Example of using a single bid to represent preferences in a Product-Mix auction. The single bid with weight 2, implementing the preferences of Ex. 1. The total demand generated by the bid is indicated in each region of price space.

Bids in which $w(\mathbf{b}) < 0$ ("negative" bids) are interpreted as "cancellation" bids that cancel part of the demand created by positive bids. But this means that all bids can be treated by the auctioneer in exactly the same way: a bid is accepted whenever at least one of its prices exceeds the corresponding auction price and, if it is accepted, then it is allocated the good on which its

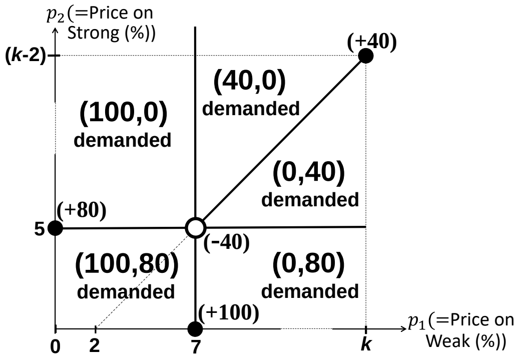

Fig. 2 Example of using positive and negative bids to represent preferences in a Product-Mix auction. The set of bids implementing the preferences of Ex. 2. The sizes of the bids ($millions) are shown next to the black and white circles that represent the positive and negative bids, respectively. The total demand generated by the complete set of bids, ($millions of weak, $millions of strong), is indicated in each region of price space.

price exceeds the corresponding auction price by most.[7] The following example from Klemperer (2008, 2010) demonstrates the potential usefulness of negative bids in the context of the Bank of England's auctions, in which the different goods were "weak collateral" and "strong collateral", and the prices were the interest rates that the winning bidders paid:[8]

Example 2. A bidder might be interested in $100 million of weak collateral (good 1) at up to a 7% interest rate, and $80 million of strong collateral (good 2) at up to a 5% interest rate. However, even if prices are high, the bidder wants an absolute minimum of $40 million (see Figure 2). These preferences can be implemented by making all of the following four bids:

I $100 million of weak at 7%.
II $80 million of strong at 5%.
III $40 million of {weak at maximum permitted bid OR strong at maximum permitted bid *less* 2%}.
IV ***minus*** $40 million of {weak at 7% OR strong at 5%}.

Note that the bids lead to an arrangement of hyperplanes, at each of which the agent is indifferent among more than one bundle. Bids (I) and (II) together generate the demand shown in the three quadrants to the left of $(7, 0)$ and/or

[7] Note that negative dot bids cannot be understood as offers to sell–an offer to sell would be accepted whenever its price is sufficiently low, whilst a negative bid cancels a purchase whenever one of its prices is sufficiently high.

[8] Although negative bids were offered as an option to the Bank of England in Klemperer (2008), its Product-Mix auctions did not use them. Prior to 2014, bidders could make any set of positive bids, and the auctioneer (the Bank of England) expressed its own preferences using a supply function that was equivalent to using any set of positive bids (see Appendix E1 of Klemperer (2018)). Since 2014, the auctions run by the Bank have allowed the auctioneer to use richer preferences than this, but have restricted to bidders to sets of bids "on the axes" (that is, to sets of bids each of which has $b_i > 0$ for only one i).

below $(0,5)$, but would on their own imply zero demand in the top right quadrant. Adding the high positive bid, (III), at $(k, k-2)$, in which k is the maximum permitted bid on either good (we assume k is large), would add demand of \$40 million of weak everywhere above the 45 deg diagonal line through $(2,0)$, and \$40 million of strong everywhere below this line; the negative bid, (IV), at $(7,5)$ then cancels this bid everywhere to the left of, and below, $(7,5)$.

Preferences of the kind illustrated in Example 2 are very natural for a liquidity-constrained bidder, but cannot be accurately represented without the use of a negative bid.[9] However, with positive and negative bids, valid bids in the bidding language can precisely represent any "strong substitutes" preferences.[10] Moreover, the way in which positive and negative bids define demand sets has a nice geometric interpretation as Minkowski differences, as we will show. And, importantly, as we discuss below, in practical settings expressing valuations with SSPMA bids is likely to be *much* more compact than listing valuations explicitly as assumed in Bikhchandani and Mamer (1997) or subsequent literature. For all these reasons, the SSPMA is a natural choice for applications.

To make practical use of SSPMAs, however, requires that we can find competitive equilibrium prices among participants using the bid language.[11] That is, given the collection of the sets of bids made by all the participants, we need to be able to find a price vector at which any given quantity vector of goods would be exactly demanded if all the bids expressed participants' actual preferences.

If all the bids are positive, the competitive equilibrium price vectors are just the shadow price vectors in the solution to a simple linear program, more specifically a network flow problem, in which the number of variables is linear in the number of bids and distinct goods. The reason is that competitive

[9] One way to understand a negative bid for a unit is that it is the highest price at which you would cancel a bid for one unit. Reducing your purchases only at *low* prices makes no sense on its own. However, in two dimensions, for example, it *does* make sense in conjunction with a positive bid north-east of the negative bid which gives higher prices at which you would buy (and that the negative bid therefore cancels when prices are low) and *also* other bids to the west and south of it, at least one of which is accepted when the cancellation operates (and without which there would be no reason for the cancellation).

[10] Klemperer (2010) stated this result for the case of multiple units of each of two goods. Baldwin and Klemperer (2016) and Baldwin and Klemperer (2021) show the general result, and also show that any preferences represented by this language that are valid (i.e., the demand for a good cannot decrease if its price falls while no other price changes–see discussion below Definition 2) must be strong substitutes.

[11] Bidders in a Product-Mix auction simultaneously make sets of bids that express their preferences. The auctioneer then chooses the aggregate supply and allocates each participant its competitive-equilibrium allocation at competitive-equilibrium prices, assuming that all the expressed preferences are accurate. Ties between bids can be broken arbitrarily, since participants who express their preferences accurately are indifferent. If there are multiple competitive equilibria, the Bank of England's Product-Mix auctions choose the best one for bidders (this is uniquely defined–see discussion below Definition 2). See Klemperer (2008, 2010) for more details.

equilibrium maximizes social surplus in our setting, so the relevant linear program allocates the bids among participants to maximize the sum of their surpluses, subject to allocating exactly the available supply.[12] With negative bids, however, the allocation problem cannot be modeled with only a single linear program, and the computation of prices is then more challenging.

1.2 Our Contribution

We study characteristics of strong substitutes by using the SSPMA language. First, we show that the positive and negative bids in the SSPMA allow us to interpret strong substitutes as Minkowski differences between sets that are easy to identify. This gives new insight into the geometric structure of strong substitutes, a valuation class that is difficult to characterize. We then illustrate the SSPMA language's expressiveness using Ostrovsky and Paes Leme (2015)'s notorious example of strong substitutes that other languages cannot represent. We also explain that the language is compact for realistic settings, since the bidder need not explicitly give a value for every bundle which it might be allocated.

Our main contribution is an equivalence result for different mathematical formulations of the pricing problem. We show that minimizing the difference between the maximum social surpluses attained by solving certain pairs of allocation problems–each of which is a simple problem–provides the information we need to compute the equilibrium prices. Specifically, the correct quantity of negative bids, s, accepted by the auctioneer minimizes the difference between the objective function of the linear program that would be solved to allocate the available supply increased by s if only the positive bids were available (we call this the "positive program"), and the objective of a corresponding linear program that would be solved to allocate a quantity of s using only the negative bids (the "negative program"). Moreover, the competitive equilibrium price vectors are the shadow price vectors for the positive program for this value of s.[13] We prove these results by showing that minimizing the difference between the positive and negative programs is dual to minimizing a Lyapunov function $L(\mathbf{p})$. More precisely, we show that the Toland-Singer dual (Maehara and Murota, 2015; Toland, 1979) of $L(\mathbf{p})$ is the minimum difference between the positive and negative linear programs.

Baldwin, Goldberg, Klemperer, and Lock (2019) have recently shown that a standard steepest-descent algorithm based on the Lyapunov function (following Murota and Tamura (2003)) can solve the SSPMA pricing problem, but their method takes only limited advantage of the special features of the geo-

[12] This is the solution method currently used by the Bank of England's Product-Mix program, which does not allow bidders to use negative bids.

[13] These shadow price vectors are a subset (often a proper subset) of the shadow price vectors for the negative program for this s.

metric representation.[14] By taking fuller advantage of the structure of strong substitutes analyzed in this paper, we find an alternative to steepest descent on the Lyapunov function. Our algorithm draws on DC (difference of convex functions) programming.[15] Steepest descent algorithms on the Lyapunov function are known to be very efficient. But we find that the DC algorithm is at least similarly fast in all our experiments. Neither algorithm is consistently faster, and in environments with only a low number of negative bids (which we conjecture are the most likely ones in practice–see Section 2.3), the DC algorithm is the faster one. So, while both algorithms terminate in a few seconds even for large problem instances, the DC algorithm provides an valuable new alternative by taking advantage of the structural properties of strong substitutes.

1.3 Outline

We proceed as follows. Section 2 introduces the SSPMA bidding language. We illustrate its expressiveness, and explain that it is a compact language that expresses all strong-substitutes valuations (and no others) as the Minkowski difference of positive and negative bids. Section 3 proves that the pricing problem can be solved by minimizing the difference between the objectives of the two linear programs, by showing that this is dual to minimizing the Lyapunov function. Section 4 takes advantage of this result to use "DC programming" (difference of convex functions programming) to specify an algorithm to solve the problem, and uses numerical experiments to compare our algorithm to a steepest-descent algorithm based on the Lyapunov function. Section 5 concludes. All proofs are in the Appendix.

2 The SSPMA Bid Language

2.1 Formal description of the SSPMA language

In the SSPMA, each of m bidders $j \in \{1, \ldots, m\}$ submit an arbitrary number of bids for distinct goods $i \in \{1, \ldots, n\}$. A bid is a vector $\mathbf{b} = (b_1, \ldots, b_n; b_{n+1}) \in \mathbb{Z}^n_{\geq 0} \times (\mathbb{Z} \setminus \{0\})$. Here, for $i = 1, \ldots, n$, coordinate b_i gives the value for good i. The final coordinate $b_{n+1} \in \mathbb{Z} \setminus \{0\}$ is the *weight* of the bid; we write $w(\mathbf{b})$ for the projection to this final coordinate. We refer to *positive* and *negative* bids according to the sign of $w(\mathbf{b})$. Prices $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^n$

[14] Unlike Baldwin, Goldberg, Klemperer, and Lock (2019) we focus on the structural properties of strong substitutes that arise from the SSPMA bid language as well as the economic interpretation of the Toland-Singer dual of the widely used Lyapunov function.

[15] Minimizing the difference between two $M^\natural$-convex functions is in general NP-hard (Kobayashi, 2015; Maehara et al., 2018): the difference between the positive and negative programs is neither convex nor concave. However, this specific problem can be solved in polynomial time, as is clear from the relationship to the Lyapunov function.

are linear. Our bundles of indivisible goods will be written $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n_{\geq 0}$. We write $\mathbf{e}^i$ for the coordinate vectors $i = 1, \ldots, n$ in $\mathbb{Z}^n$.

A positive bid $\mathbf{b}$ expresses the willingness of the bidder to pay at most b_i for units of good $i = 1, \ldots, n$, and for up to $w(\mathbf{b})$ units in total. It defines a *valuation* $v_{\mathbf{b}}$ on the domain $\Delta_{w(\mathbf{b})}$ of bundles of at most $w(\mathbf{b})$ units, that is, $\Delta_{w(\mathbf{b})} = \{\mathbf{y} \in \mathbb{Z}^n_{\geq 0} : \sum_{i=1}^n y_i \leq w(\mathbf{b})\}$, with $v_{\mathbf{b}}(\mathbf{y}) = \sum_{i=1}^n b_i y_i$. The utility associated with this bid is quasi-linear, $v_{\mathbf{b}}(\mathbf{y}) - \langle \mathbf{p}, \mathbf{y} \rangle$, so the *indirect utility* associated with such a bid is just

$$u_{\mathbf{b}}(\mathbf{p}) = w(\mathbf{b}) \max_{i \in \{1,\ldots,n\}} (b_i - p_i, 0), \tag{1}$$

where we include 0 because the bid may instead be rejected. Any combination of $w(\mathbf{b})$ units of utility-maximizing goods is acceptable, as are fewer units when $u_{\mathbf{b}}(\mathbf{p}) = 0$, so the *demand set* is

$$D_{\mathbf{b}}(\mathbf{p}) := \left\{ \mathbf{y} \in \Delta_{w(\mathbf{b})} \ : \ \sum_{i=1}^n y_i(b_i - p_i) = u_{\mathbf{b}}(\mathbf{p}) \right\}. \tag{2}$$

This set comprises all integer bundles in the convex polytope in which the bundles $w(\mathbf{b})\mathbf{e}^i$, where i maximizes $b_i - p_i \geq 0$, are vertices, and $\mathbf{0}$ is also a vertex if $\max_{i \in \{1,\ldots,n\}}(b_i - p_i, 0) = 0$. If $D_{\mathbf{b}}(\mathbf{p})$ contains more than one bundle, we say all goods $i = 1, \ldots, n$ maximizing $b_i - p_i$ are *marginal* for bid $\mathbf{b}$ at $\mathbf{p}$. If $\{\mathbf{0}\} \subsetneq D_{\mathbf{b}}(\mathbf{p})$ then we say the bid is *marginal to be accepted.* If $D_{\mathbf{b}}(\mathbf{p}) = \{\mathbf{0}\}$ we say the bid is *rejected.*

Now consider a multiset $\mathcal{B}$ of positive bids, which could be all those placed by a single bidder, or could, for example, be all bids from *all* bidders. The aggregate indirect utility $u_{\mathcal{B}}(\mathbf{p})$ is just the sum of indirect utilities: $u_{\mathcal{B}}(\mathbf{p}) = \sum_{\mathbf{b} \in \mathcal{B}} u_{\mathbf{b}}(\mathbf{p})$, and the aggregate demand set $D_{\mathcal{B}}(\mathbf{p})$ is the Minkowski sum of demand sets $D_{\mathcal{B}}(\mathbf{p}) = \sum_{\mathbf{b} \in \mathcal{B}} D_{\mathbf{b}}(\mathbf{p})$.

However, we also allow negative bids: those for which $w(\mathbf{b}) < 0$. These do not represent a meaningful economic valuation on their own, but do so in "valid" combinations with positive bids. Given a collection $\mathcal{B}$ of bids, write respectively $\mathcal{B}_+$ and $\mathcal{B}_-$ for the positive and negative bids in $\mathcal{B}$. Write $|\mathbf{b}|$ for the bid $(b_1, \ldots, b_n; |w(\mathbf{b})|)$, and write $|\mathcal{B}_-|$ for the set of bids $|\mathbf{b}|$ where $\mathbf{b} \in \mathcal{B}_-$. Now the *aggregate indirect utility* is an appropriately signed sum of indirect utilities:

$$u_{\mathcal{B}}(\mathbf{p}) := \sum_{\mathbf{b} \in \mathcal{B}_+} u_{\mathbf{b}}(\mathbf{p}) - \sum_{\mathbf{b} \in |\mathcal{B}_-|} u_{\mathbf{b}}(\mathbf{p}). \tag{3}$$

We say that the set $\mathcal{B}$ is *valid* when the indirect utility $u_{\mathcal{B}}$ is concave. (See Theorem 1 of Baldwin, Goldberg, Klemperer, and Lock (2019); further discussion of this notion is given below after Definition 2.)

To define the *aggregate demand set* with positive and negative bids, first define the demand $D_{\mathbf{b}}(\mathbf{p})$ associated with an individual negative bid $\mathbf{b}$ as $D_{\mathbf{b}}(\mathbf{p}) = -D_{|\mathbf{b}|}(\mathbf{p}) = \{-\mathbf{x} \mid \mathbf{x} \in D_{|\mathbf{b}|}(\mathbf{p})\}$. Let $\mathcal{Q}$ comprise all price vectors $\mathbf{q}$ in a small neighborhood of $\mathbf{p}$, and such that $D_{\mathbf{b}}(\mathbf{q}) = \{\mathbf{x}_{\mathbf{b}}(\mathbf{q})\}$ are singletons for all $\mathbf{b} \in \mathcal{B}$. Then the aggregate demand set is equal to the discrete convex hull

$$D_{\mathcal{B}}(\mathbf{p}) = \operatorname{conv}\left\{ \sum_{\mathbf{b}\in\mathcal{B}} D_{\mathbf{b}}(\mathbf{q}) : \mathbf{q} \in \mathcal{Q} \right\} \cap \mathbb{Z}^n.$$

In particular, if $D_{\mathbf{b}}(\mathbf{p})$ is a singleton for all $\mathbf{b} \in \mathcal{B}$, then $D_{\mathcal{B}}(\mathbf{p})$ is just $\sum_{\mathbf{b}\in\mathcal{B}} D_{\mathbf{b}}(\mathbf{p}) = \sum_{\mathbf{b}\in\mathcal{B}_+} D_{\mathbf{b}}(\mathbf{p}) - \sum_{\mathbf{b}\in|\mathcal{B}_-|} D_{\mathbf{b}}(\mathbf{p})$: negative bids are used to "cancel" part of the demand arising from positive bids. We cannot extend this rule to prices at which the demand set is non-unique simply by taking the Minkowski sum of demand sets associated with all bids; negative bids which are marginal between goods must be treated consistently with positive bids marginal on those same goods.[16] However, if the bids $\mathcal{B}^j$ of each bidder $j = 1, \ldots, m$ are valid, then the full aggregate demand set $D_{\mathcal{B}}(\mathbf{p})$ defined by $\mathcal{B} = \bigcup_{j=1}^m \mathcal{B}^j$ is indeed the Minkowski sum: $D_{\mathcal{B}}(\mathbf{p}) = \sum_{j=1}^m D_{\mathcal{B}^j}(\mathbf{p})$.

When $\mathcal{B}$ contains only positive bids, we can aggregate the simple valuations implied by individual bids, to obtain the *aggregate valuation* $v_{\mathcal{B}} : \varDelta_W \to \mathbb{Z}$, where $W = \sum_{\mathbf{b}\in\mathcal{B}} w(\mathbf{b})$:

$$v_{\mathcal{B}}(\mathbf{y}) = \max\left\{ \sum_{\mathbf{b}\in\mathcal{B}} \sum_{i=1}^n x_{i\mathbf{b}} b_i : \sum_{\mathbf{b}\in\mathcal{B}} x_{i\mathbf{b}} \leq y_i \,\forall i \text{ and } \sum_{i=1}^n x_{i\mathbf{b}} \leq w(\mathbf{b}) \,\forall \mathbf{b} \in \mathcal{B} \right\}. \tag{4}$$

As usual, the relations $u_{\mathcal{B}}(\mathbf{p}) = \max_{\mathbf{x}\in\mathbb{Z}^n_{\geq 0}} v_{\mathcal{B}}(\mathbf{x}) - \langle \mathbf{p}, \mathbf{x} \rangle$ and $v_{\mathcal{B}}(\mathbf{x}) = \min_{\mathbf{p}\in\mathbb{R}^n} u_{\mathcal{B}}(\mathbf{p}) + \langle \mathbf{p}, \mathbf{x} \rangle$ hold. The latter equation also gives us an indirect way to identify the aggregate valuation if $\mathcal{B}$ is a valid set of positive and negative bids. However, one of our main results, which is also the starting point to our equilibrium pricing algorithm, is a purely primal expression for the aggregate valuation in the presence of negative bids (Theorem 1).

The valuation implied by such bids is for strong substitutes:

Definition 1 (Ordinary and Strong Substitutes, Milgrom and Strulovici (2009) and Baldwin and Klemperer (2019)) A valuation v is *ordinary substitutes*, if for any price vectors $\mathbf{p}' \geq \mathbf{p}$ with singleton demand sets $D_v(\mathbf{p}') = \{\mathbf{x}'\}$ and $D_v(\mathbf{p}) = \{\mathbf{x}\}$, we have $\mathbf{x}'_k \geq \mathbf{x}_k$ for all k with $\mathbf{p}'_k = \mathbf{p}_k$. A valuation v is *strong substitutes*, if, when we consider every unit of every good to be a separate good, v is ordinary substitutes.

The SSPMA only expresses preferences of this kind, and can express *any* strong substitutes valuation (Baldwin and Klemperer, 2016, 2021)[17]. It is, to our knowledge, the only bidding language that provably has this feature.

In an SSPMA such as the Bank of England's, total supply is not predetermined; the auctioneer represents its preferences as supply schedules, and

[16] For example, $(140, 40)$ is *not* in the demand set at $\mathbf{p} = (3, 1)$ in the right-hand side of Figure 1; the bids for -40 and 40 units must be treated consistently.

[17] Klemperer (2010) stated this result for the case of multiple units of two goods. Lin and Tran (2017) describes how any valuation can be analyzed tropical-geometrically and can be decomposed into a combination of simpler pieces, but if the valuation is not strong substitutes, these simpler pieces do not correspond to positive and negative bids.

the auction finds competitive equilibrium given the auctioneer's and bidders' expressed preferences. However, the auctioneer can equivalently auction the maximum quantity of each good that it would ever sell at any price vector, and place appropriate bids to buy back quantities at lower prices.[18] So we index the auctioneer as agent 0 and include its bids in the set $\mathcal{B}$ of all bids from all bidders. This paper therefore addresses the following problem:

Definition 2 (Equilibrium pricing problem) Given a valid set $\mathcal{B}$ of all bids from all bidders (including the auctioneer) and a *target supply* $\mathbf{t}$, find a price vector $\mathbf{p} \in \mathbb{R}^n$, such that $\mathbf{t}$ is demanded at $\mathbf{p}$, that is, $\mathbf{t} \in D_{\mathcal{B}}(\mathbf{p})$. Such a price vector is called an equilibrium price.[19]

It is well-known that a competitive equilibrium does indeed exist, given our assumptions of strong substitutes and a seller who will retain units of any underdemanded good at a price of zero (Danilov et al., 2001; Milgrom and Strulovici, 2009). Indeed, this also implies that equilibrium price in $\mathbb{R}^n_{\geq 0}$ exists. The set of equilibrium prices forms a lattice with respect to the Euclidean ordering (Gul and Stacchetti, 1999; Murota, 2003), i.e., for any valuations $v_1, \cdots, v_m$, if $\mathbf{p}$ and $\mathbf{p}'$ are equilibrium prices for such valuations, then $\mathbf{p} \wedge \mathbf{p}'$ and $\mathbf{p} \vee \mathbf{p}'$ are also equilibrium prices. This implies that there exists an unique minimal equilibrium price vector. It is possible to modify the algorithm we will develop to find the minimal equilibrium price vector rather than an arbitrary price vector.

To understand the "validity" of bids in the SSPMA, we briefly outline some geometric ideas from Baldwin and Klemperer (2019). First, the collection $\mathcal{B}$ of bids induces a set of prices at which the aggregate demand is not unique: the "locus of indifference prices" (LIP), notated $\mathcal{L}_{\mathcal{B}} := \{\mathbf{p} : |D_{\mathcal{B}}(\mathbf{p})| > 1\}$. For a price $\mathbf{p}$ to be in the LIP, at least one bid must be marginal, so some equality of the form $b_i = p_i$ or $b_i - p_i = b_j - p_j$ must hold, where $i, j \in \{1, \ldots, n\}$ and $j \neq i$. Therefore, $\mathcal{L}_{\mathcal{B}}$ consists of a union of pieces of hyperplanes with normals in $\{\mathbf{e}^i, \mathbf{e}^i - \mathbf{e}^j : 1 \leq i < j \leq n\}$. These pieces of hyperplanes are known as *facets*. To each facet F, we assign a *weight* $w(F)$, given by the sum of the weights of bids that are marginal at a price in the relative interior of F. Facets always have nonzero weight; if the sum of weights of marginal bids is zero then one may see that demand is in fact unique.

The LIP $\mathcal{L}_{\mathcal{B}}$ splits price space into multiple *unique demand regions (UDRs)* at which a unique bundle is demanded. Let $\mathbf{p}$ be a price vector in an UDR for which the demand is known (for example, for $\mathbf{p}$ large, the demand is 0). If the price $\mathbf{p}$ changes along a curve, and crosses a facet F of $\mathcal{L}_{\mathcal{B}}$, then the demand

[18] Klemperer (2018) (Appendix E1) illustrates how the auctioneer can do this for general supply schedules; for the special case in which it just wishes to sell a bundle $\mathbf{t} = (t_1, \ldots, t_n)$ at any non-negative prices, it will simply auction supply $\mathbf{t}$ and enter a bid $(0, \ldots, 0; \sum_i t_i)$ into the auction.

[19] This paper takes competitive behavior as given; we do not address the extent to which bidders may distort their preferences. In an SSPMA it is rational for bidders whose demand is not too large relative to aggregate demand to make bids that approximately reflect their true preferences. Throughout this article, we assume that bids reflect true preferences.

changes by $w(F)\mathbf{n}$, where $\mathbf{n}$ is the normal of F pointing into the opposite direction of the path. For an illustration, see Figure 3. Thus, the LIP fully encodes the aggregate demand at every UDR-price, and so – by taking convex hulls – at every price.

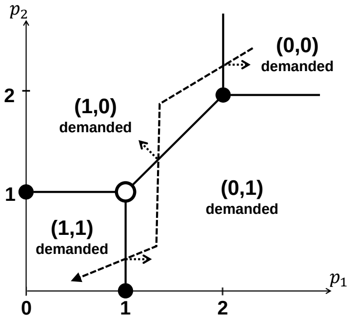

Fig. 3 Finding demand in each UDR of a LIP. The black circles represent positive bids with weight 1, namely $(2,2;1),(1,0;1)$ and $(0,1;1)$; the white circle represents a negative bid, $(1,1;-1)$. Note that all facets emanating from this negative bid are canceled by parts of facets arising from positive bids. A curve which determines demand in every UDR is shown as a dashed line. The curve starts at a high price, where the demand is $(0,0)$. The vectors where the path intersects the LIP indicate the correctly oriented normals of the facets with respect to the path. For example, inspecting the crossings of facets reveals that the demand at $(0.5,0.5)$ is $1\cdot(1,0)+1\cdot(-1,1)+1\cdot(1,0)=(1,1)$.

Now, a negative-weighted facet cannot arise from a quasi-linear preference relation: when the price of one good decreases, the demand for that good must not also decrease. So negative bids must be placed in such a way that, in the resulting collection of facets, no facet has a negative weight. This condition is equivalent to concavity of the indirect utility function.[20] From now on we assume that *our bid collections are always valid.* Note that if each individual bidder's bid set is valid, then so is the set of all bids from all bidders.

The following two subsections discuss geometrical interpretations and properties of this bid language. While they do not contribute directly to our main Theorem 1 and the algorithm, they provide useful background on the role of negative bids in the SSPMA and intuition for the overall approach.

2.2 Interpretation via Minkowski Differences

There appears to be a contrast between the intuitive definition of the aggregate demand set when all bids are positive (so the aggregate demand set is just the Minkowski sum of the individual demand sets) and the more involved definition

[20] See Baldwin, Goldberg, Klemperer, and Lock (2019) Theorem 1, which shows that, at any price, the sum of the weights of bids marginal between any pair of goods, or between any good and being rejected, must be non-negative. The failure of this condition is equivalent to the existence of a negative-weighted facet.

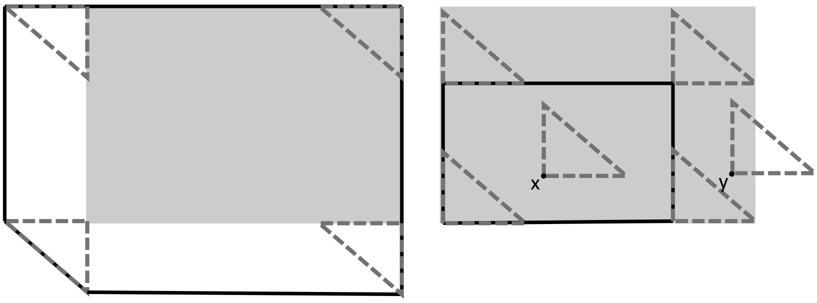

Fig. 4 The Minkowski sum $A+(-B)$ and Minkowski difference $A-B$ of a rectangle A and a triangle B. Left: the rectangle A (in gray); four instances of $a + (-B)$, in which $a \in A$ and $-B$ is the triangle (dashed line and its interior); and $A + (-B)$ (black line, including its interior). Right: the same rectangle A (in gray); six instances of $a + B$. For five of these, such as $a = x$, we have $a + B \subseteq A$ and so $a \in A - B$, but $y + B \not\subseteq B$ and so $y \notin A - B$. The full set $A - B$ is given by the black line and its interior.

when negative bids are present. Recall from Section 2.1 that in this case we defined the aggregate demand set to be the discrete convex hull of bundles which are demanded uniquely when we slightly change the price vector $\mathbf{p}$. We cannot simply take Minkowski sums because we must ensure that negative bids are treated in a valid way with their associated positive bids (see the discussion after Definition 2). However, if $\mathcal{B}$ is valid, then we can provide a more parsimonious novel definition by using the Minkowski difference operation. We recall that $A - B$ consists of all points $\mathbf{x} \in \mathbb{R}^n$, such that $\mathbf{x} + B \subseteq A$. The geometric effect of this operation is illustrated in Figure 4. Note in particular that in general $A + (-B) \neq A - B$.

Proposition 1 *Let $\mathcal{B}$ be a valid collection of bids in an SSPMA. Then for every price vector $\mathbf{p}$ the demand set $D_{\mathcal{B}}(\mathbf{p})$ is equal to $D_{\mathcal{B}_+}(\mathbf{p}) - D_{|\mathcal{B}_-|}(\mathbf{p})$.*

We prove this result in Appendix A.1.

2.3 Expressiveness and compactness of the SSPMA

To illustrate the expressive power of negative bids, we consider Ostrovsky and Paes Leme (2015)'s notorious example of a valuation, v_r, that shows that prior bid languages such as the endowed assignment valuations by Hatfield and Milgrom (2005) are strictly less expressive than the set of gross substitutes (and so also strong substitutes) valuations.[21] (We discuss the construction of v_r in Appendix A.2.) However,

Proposition 2 *The valuation v_r in Ostrovsky and Paes Leme (2015) can be represented by 8 positive and 6 negative SSPMA bids.*

[21] Fichtl (2021) uses the same example to show Milgrom (2009)'s (integer) assignment messages cannot express all strong substitute valuations.

This proposition illustrates Baldwin and Klemperer (2021)'s more general result that *all* strong substitutes valuations can be depicted in the SSPMA. We prove Proposition 2 without relying on this general result by explicitly providing the list of SSPMA bids.

Moreover, an important feature of the SSPMA language is that it is parsimonious: the valuations that are most used in practice can be expressed very simply, using far fewer bids than the number of different bundles valued. Let $W := \sum_{\mathbf{b}\in\mathcal{B}} w(\mathbf{b})$. Note that W equals the maximum number of units that a bidder who makes bids $\mathbf{b} \in \mathcal{B}$ is interested in. Then SSPMA bids can assign a "non-trivial" value to $\Omega(W^n)$ bundles:

Proposition 3 *Consider an SSPMA with n goods, and suppose a bidder makes bids $\mathcal{B}$. Let $D := \bigcup\{D_{\mathcal{B}}(\mathbf{p}) : \mathbf{p} \in \mathbb{R}^n\}$ and let $W := \sum_{\mathbf{b}\in\mathcal{B}} w(\mathbf{b})$. Then $D = \Delta_W$ and so $|D| = \binom{n+W}{n} \geq (1 + W/n)^n$.*

Moreover, bidders in practical applications are likely to need to make far fewer bids than Proposition 3 suggests: Expressing a demand function for each good independently is trivial—it just requires providing a separate list of bids for each i with, for each i, $b_j = 0$ for all $j \neq i$. In many settings these bids will express much of the information about bidders' valuations.

At a second, higher, level of complexity, any bid which selects the "best value" among any number of goods can be expressed using only positive bids. Observe that W is the maximum number of bids that a bidder who is interested in winning at most W units, and who uses only positive bids, needs to make—and if any of her bids have greater weight than 1, she will need fewer bids. So such a bidder can express her valuations of all possible bundles with only a few bids.

More complex features of preferences require negative bids to express, but these features seem less likely to arise frequently. Example 2 is one example, and there are others,[22] but we expect most bidders would be unlikely to have to handle more than a very small number of these special issues. In fact, in the Bank of England's auctions, bidders showed relatively little interest even in bids of the "second level" of complexity, and they used such bids only rarely—perhaps because they are only very important to banks in times of real crisis.[23] So bidders are unlikely to need many negative bids in most practical

[22] For example, choosing the best N_1 out of N_2 options requires the use of negative bids if $N_2 > N_1 > 1$.

[23] The need for "second-level" bids is likely to grow, however, as technology develops– they are most useful for banks who can coordinate different parts of their operations in a sophisticated way, and "big investment programmes are already underway in many [banks], to ensure that [they] have real-time information on the collateral they have available globally across all their business lines, that the collateral they deliver is cost effective, and that the cost of delivering (and financing) that collateral is factored into their risk and business decisions. These programmes involve sometimes relatively advanced technology; indeed, as some of our contacts remark, somewhat alarmed, 'for the first time in living memory, pointy heads are sitting on the repo desk'." (Andrew Hauser [Executive Director of the Bank of England], 2013) (Hauser, 2013).

auction settings.[24] The number of bids needed by a bidder who is interested in winning at most W units, and who needs only a small number of negative bids, cannot much exceed W. Moreover, such a bidder will need to use many fewer than W bids unless most or all of her bids are of weight only 1. So these bidders, too, are likely to be able to express their full valuations with only a few bids.

The number of bundles valued by a set of bids of mixed sign can be much smaller than the lower bound on the number of bundles valued by the same number of bids that are all positive.[25] But we expect the SSPMA bidding language will be much more "compact"[26] in most practical cases than—say—listing valuations for all bundles explicitly.

So in the settings that we believe are most likely to arise, bidders can probably build up their valuations bit by bit through adding individual bids. We expect most of these bids will be positive; moreover, software can make entering negative bids easier, by checking feasibility and by allowing the bidding of "words,"[27], and by checking feasibility.[28]

At least with human participants, it seems less likely that a valuation function with all exponentially many package values would be available, but in this case the prices can be computed directly, using steepest descent or linear programming algorithms. Alternatively, bidders could use Goldberg et al. (2020)'s algorithm to generate bids from an arbitrary value function.[29]

3 The SSPMA Pricing Problem

With only positive bids, our equilibrium pricing problem (Definition 2) can be solved via a simple linear program that maximizes the total welfare given the target bundle $\mathbf{t}$. We know that $\mathbf{t} \in \varDelta_W$, where W is the total weight of

[24] Moreover, Klemperer (2018) shows how to enhance the SSPMA with additional "words", each of which refers to a particular configuration of positive and negative bids. This can greatly reduce the number of bids required to express special situations. For example, our Example 2 could be expressed by a single "word" from a parameterised class of words. Preferences of the kind described in note 22 could also be expressed as "words".

[25] Of course, no language can express every possible valuation using fewer pieces of information than the number of bundles that can be independently valued. However, in extreme cases the number of bids required to express a full valuation for up to W units in the SSPMA can exceed the number of different possible bundles of up to W units.

[26] See Goetzendorff et al. (2015) for a discussion of compactness.

[27] See note 24.

[28] In more complex cases, we can use the Goldberg et al. (2020) algorithm to generate bids from an arbitrary value function.

[29] Goldberg et al. (2020)'s algorithm has linear query complexity for preferences that require positive bids only. (An asymptotic lower bound of $\varOmega(B \log M)$ queries are required to learn a list of B positive bids, where M is the magnitude of the bid vectors w.r.t. the L_∞-norm.) It has exponential query complexity in the worst case when negative bids are required. (The query complexity of learning bid lists corresponding to strong substitutes demand has a rate of growth of $\varTheta(B \log M + B^n)$.) However, if the number of goods is not too large, the algorithm still performs well, even though Lin and Tran (2017) observe that breaking a general valuation up into constituent simpler parts can be NP-hard.

bids placed, by our assumption about the bids of the auctioneer. But recall from Equation (4) that, given the collection $\mathcal{B}$ of *all* bids of *all* bidders, the aggregate valuation of any bundle $\mathbf{t} \in \Delta_W$ is given - in LP notation - by

$$\begin{aligned} v_{\mathcal{B}}(\mathbf{t}) = \max \sum_{\mathbf{b}\in\mathcal{B}} \sum_{i\in[n]} & b_i x_{\mathbf{b}i} && && \text{(LP)} \\ \text{s.t.} \sum_{i\in[n]} & x_{\mathbf{b}i} \le w(\mathbf{b}) && \forall \mathbf{b} \in \mathcal{B} && (\pi_{\mathbf{b}}) \\ \sum_{\mathbf{b}\in\mathcal{B}} & x_{\mathbf{b}i} = t_i && \forall i \in [n] && (p_i) \\ & x_{\mathbf{b}i} \ge 0 && \forall \mathbf{b} \in \mathcal{B},\, i \in [n]. \end{aligned}$$

Here $\pi_{\mathbf{b}}$ and p_i denote the respective dual variables. This program always has an integral optimal solution, as may be seen either by properties of strong-substitutes valuations, or by recognizing that it is an instance of the min-cost flow problem. The number of constraints and variables is polynomial in the number of bids and goods, in contrast to the formulation of Bikhchandani and Mamer (1997). The set of equilibrium prices can be computed directly:

Proposition 4 *In an SSPMA with only positive bids, the equilibrium prices for the target supply* $\mathbf{t}$ *are the optimal dual variables* $\mathbf{p} = (p_1, \ldots, p_n)$ *of the network linear program (LP) which can be solved in polynomial time in the number of goods and bids.*

Proposition 4 simply follows from writing down the complementary slackness conditions of (LP), so we do not provide an explicit proof. If $\mathcal{B}$ also contains negative bids, the problem of efficiently computing equilibrium prices is less obvious. One route, taken by Baldwin, Goldberg, Klemperer, and Lock (2019), is to minimize the *Lyapunov function* $L : \mathbb{R}^n \to \mathbb{R}$ (Ausubel, 2006), defined for target $\mathbf{t}$ as

$$L(\mathbf{p}) = u_{\mathcal{B}}(\mathbf{p}) + \langle \mathbf{p}, \mathbf{t} \rangle$$

where aggregate indirect utility $u_{\mathcal{B}}(\mathbf{p})$ is as defined in Equation (3). The set of minimizers of L coincides with the set of equilibrium prices, and structural properties of L allow for polynomial-time steepest descent algorithms to find these minima (Baldwin et al., 2019; Murota, 2003; Paes Leme and Wong, 2017). However, this approach works by invoking a rather generic submodular function minimization algorithm, under the assumption that a demand oracle is available.

By contrast, with only positive bids we can build upon much more specialized algorithms to solve network linear programs. And, as we now show, taking advantage of the economic structure of the problem allows us to incorporate negative bids into this approach:

Recall that the total allocation in an SSPMA is equal to that assigned to positive bids *minus* that assigned to negative bids. So, to assign $\mathbf{t}$ units in total, we must assign $\mathbf{t} + \mathbf{s}$ units to positive bids and $\mathbf{s}$ to negative bids, for some “supplementary” bundle $\mathbf{s}$. Recall also that we write $\mathcal{B}_+$ for the positive bids in $\mathcal{B}$, and $|\mathcal{B}_-|$ for the negative bids $\mathbf{b} \in \mathcal{B}$ endowed with weights $|w(\mathbf{b})|$. We introduce two additional SSPMAs: that with bids $\mathcal{B}_+$ and target $\mathbf{t} + \mathbf{s}$, which we call the "positive auction"; and that with (positive) bids $|\mathcal{B}_-|$ and target $\mathbf{s}$, which we call the "negative auction". Write W_+ and W_- for the total weights of bids in these respective auctions, so that Δ_{W_+} and Δ_{W_-} are the sets of bundles that may be sold by each of them. Note that for each $\mathbf{t} \in \Delta_W$ and $\mathbf{s} \in \Delta_{W_-}$, $\mathbf{t} + \mathbf{s}$ lies in Δ_{W_+} (see Appendix Lemma 4).

If we pick $\mathbf{s}$ correctly, then this is equivalent to allocating $\mathbf{t}$ units in the auction with bids $\mathcal{B}$. Moreover, since both $\mathcal{B}_+$ and $|\mathcal{B}_-|$ are sets of positive bids, their respective aggregate valuations and equilibrium prices can be evaluated using the linear program above. We now show how to find $\mathbf{s}$:

Theorem 1 *If $\mathcal{B}$ represents all bids from all bidders, then the aggregate valuation at the target supply $\mathbf{t} \in \Delta_W$ can be written as*

$$v_{\mathcal{B}}(\mathbf{t}) = \min_{\mathbf{s} \in \Delta_{W_-}} \left(v_{\mathcal{B}_+}(\mathbf{t} + \mathbf{s}) - v_{|\mathcal{B}_-|}(\mathbf{s}) \right).$$

Moreover, given a minimizer $\bar{\mathbf{s}}$, each equilibrium price $\bar{\mathbf{p}}$ of the auction with bids $\mathcal{B}_+$ and target supply $\mathbf{t} + \bar{\mathbf{s}}$ is an equilibrium price for the auction with bids $|\mathcal{B}_-|$ and target supply $\bar{\mathbf{s}}$, and also for the complete auction with bids $\mathcal{B}$ and target supply $\mathbf{t}$.

We prove Theorem 1 in Appendix A.4.

To understand the economic intuition underlying Theorem 1 assume that the set of equilibrium prices is n-dimensional and consider a price $\mathbf{p}$ in its interior. (Although the SSPMA would choose the minimum of the equilibrium prices, choosing an interior price simplifies the intuition.) Let $\bar{\mathbf{s}}$ be the vector of negative bids accepted in the equilibrium. Initially set the target $\mathbf{s}$ of the negative auction to be $\bar{\mathbf{s}}$, which means that $\mathbf{p}$ is an equilibrium price for both the positive and negative auctions.

Consider the effect of changing $\mathbf{p}$ on the weighted sum of bids accepted in these two auctions. Recall that the full set $\mathcal{B}$ of positive and negative bids in the original SSPMA is valid. So for any price at which additional negative bids are marginal to be accepted, positive bids with at least as great a weight must also be marginal to be accepted–see the discussion of validity of $\mathcal{B}$ at the end of Section 2.1. (The converse does not hold: positive bids can be marginal at prices at which no negative bid is marginal.) So, any change in price from $\mathbf{p}$ would alter the total weight of bids accepted in the positive auction by weakly more than it would alter the total weight of bids accepted in the negative auction.

Now consider an increase in one coordinate of the supplementary bundle, from $\bar{\mathbf{s}}$ to $\mathbf{s} \geq \bar{\mathbf{s}}$, in both the positive and negative auctions. The additional bids that will be accepted in the positive auction with target $\mathbf{t}+\mathbf{s}$ will, because of our observation above, have weakly greater value than the additional bids accepted in the negative auction. That is, $v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - v_{\mathcal{B}_+}(\mathbf{t}+\bar{\mathbf{s}}) \geq v_{|\mathcal{B}_-|}(\mathbf{s}) - v_{|\mathcal{B}_-|}(\bar{\mathbf{s}})$. Similarly, if we decrease one coordinate to $\mathbf{s} \leq \bar{\mathbf{s}}$, then bids which are now rejected from the positive auction will have weakly lower value than the bids rejected from the negative auction. So, again, $v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - v_{\mathcal{B}_+}(\mathbf{t}+\bar{\mathbf{s}}) \geq v_{|\mathcal{B}_-|}(\mathbf{s}) - v_{|\mathcal{B}_-|}(\bar{\mathbf{s}})$. General changes in $\mathbf{s}$ may be understood as a sequence of these two operations.

It follows that $\bar{\mathbf{s}}$ can be identified by minimizing $v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - v_{|\mathcal{B}_-|}(\mathbf{s})$.

The formal proof of Theorem 1 rests on applying a version of Toland-Singer duality (Toland, 1979) to the valuations in the positive and negative auctions, and relating this to the Lyapunov function $L(\mathbf{p})$. Maehara and Murota (2015) provide a theoretical treatment of discrete DC-functions, establishing (their Theorem 4.6) Toland-Singer duality in discrete DC-functions.

First recall that, for a function $f : \operatorname{dom} f \to \mathbb{R}$, where $\operatorname{dom} f \subseteq \mathbb{R}^n$, the convex conjugate $f^* : \operatorname{dom} f^* \to \mathbb{R}$ is defined by $f^*(\mathbf{p}) = \sup_{\mathbf{x}\in\operatorname{dom} f}(\langle \mathbf{p}, \mathbf{x}\rangle - f(\mathbf{x}))$, where $\operatorname{dom} f^* \subseteq \mathbb{R}^n$ comprises those $\mathbf{p}$ at which $f^*(\mathbf{p})$ is finite-valued. The *subdifferential* of f is the set-valued function

$$\partial f(\mathbf{x}) = \{\mathbf{p} \in \mathbb{R}^n : \langle \mathbf{p}, \mathbf{y}\rangle - f(\mathbf{y}) \leq \langle \mathbf{p}, \mathbf{x}\rangle - f(\mathbf{x}) \quad \forall \mathbf{y} \in \mathbb{R}^n\}.$$

The *domain* $\operatorname{dom} \partial f$ of the subdifferential consists of all points $\mathbf{x} \in \operatorname{dom} f$ with $\partial f(\mathbf{x}) \neq \emptyset$. It turns out that for in our application, the convex conjugates and subdifferentials have an intuitive economic meaning.

Lemma 1 *Let $\mathcal{B}$ be a collection of positive bids. Then $-v_{\mathcal{B}}$ can be naturally extended to a convex function $f : \operatorname{dom} f \to \mathbb{R}$ with the following properties:*

1. $\operatorname{dom} \partial f = \operatorname{dom} f = \operatorname{conv} \Delta_W$ *and* $\operatorname{dom} \partial f^* = \operatorname{dom} f^* = \mathbb{R}^n$
2. $f^*(\mathbf{q}) = u_{\mathcal{B}}(-\mathbf{q})$ *and* $\partial f^*(\mathbf{q}) = \operatorname{conv} D_{\mathcal{B}}(-\mathbf{q})$
3. $\partial f(\mathbf{x}) = -\{\mathbf{p} \in \mathbb{R}^n : \mathbf{x} \in \operatorname{conv} D_{\mathcal{B}}(\mathbf{p})\}$.

We will use the following version of Toland-Singer duality, which allows for restricted domains:

Theorem 2 (Toland-Singer Duality) *Let $f : \operatorname{dom} f \to \mathbb{R}$ and $g : \operatorname{dom} g \to \mathbb{R}$ be proper convex lower semi-continuous functions with closed and convex domains $\operatorname{dom} f \subseteq \operatorname{dom} g \subseteq \mathbb{R}^n$ and such that $\operatorname{dom} g^* \subseteq \operatorname{dom} f^* \subseteq \mathbb{R}^n$. If one of the differences $f(\mathbf{x}) - g(\mathbf{x})$ and $g^*(\mathbf{y}) - f^*(\mathbf{y})$ has a minimum in $\operatorname{dom} f$, respectively $\operatorname{dom} g^*$, the other difference also has one, and*

$$\min_{\mathbf{x}\in\operatorname{dom} f} f(\mathbf{x}) - g(\mathbf{x}) = \min_{\mathbf{y}\in\operatorname{dom} g^*} g^*(\mathbf{y}) - f^*(\mathbf{y}).$$

Moreover, if $\bar{\mathbf{x}}$ minimizes $f(\mathbf{x}) - g(\mathbf{x})$, then any $\bar{\mathbf{y}} \in \partial g(\bar{\mathbf{x}})$ minimizes $g^*(\bar{\mathbf{y}}) - f^*(\bar{\mathbf{y}})$. Conversely, for any minimizer $\bar{\mathbf{y}}$ of $g^*(\mathbf{y}) - f^*(\mathbf{y})$, any $\bar{\mathbf{x}} \in \partial f^*(\bar{\mathbf{y}})$ minimizes $f(\mathbf{x}) - g(\mathbf{x})$.

For a proof see Tao and An (1997, Theorem 1). We will apply Theorem 2 to the convex extensions of $-v_{|\mathcal{B}_-|}$ and $-v_{\mathcal{B}_+}$ (to the convex hulls of their domains).

4 The Pricing Algorithm

Using Theorem 1, we can approach the pricing problem by minimizing the difference $v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - v_{|\mathcal{B}_-|}(\mathbf{s})$. While the valuations $v_{\mathcal{B}_+}$ and $v_{|\mathcal{B}_-|}$ can be extended to concave functions, and can efficiently be evaluated with linear programs at any given pair of bundles, their difference is in general neither concave nor convex. Moreover, as recently shown by Kobayashi (2015), minimizing the difference between two $M^\natural$-convex functions is an NP-hard optimization problem. However, there is a class of algorithms on the difference of convex functions (DC algorithms; see An and Tao, 2005; Tao and An, 1997), that find at least local minima of such problems and are often very fast in practice.

4.1 A DC Auction Algorithm

By Theorem 1, we seek $\bar{\mathbf{s}}$ minimizing $v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - v_{|\mathcal{B}_-|}(\mathbf{s})$. We will approach this by minimizing $f(\mathbf{s}) - g(\mathbf{s})$, where $f(\mathbf{s})$ and $g(\mathbf{s})$ are the convex extensions of $-v_{|\mathcal{B}_-|}(\mathbf{s})$, respectively $-v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s})$, to the convex hulls of their domains. A *necessary* condition for such $\bar{\mathbf{s}}$ is that it gives a stationary point, that is, $\bar{\mathbf{s}} \in \operatorname{dom} \partial f$ with $\partial f(\bar{s}) \cap \partial g(\bar{s}) \neq \emptyset$. To interpret this in our context, if $\mathbf{q} \in \partial f(\bar{\mathbf{s}}) \cap \partial g(\bar{\mathbf{s}})$ then $\mathbf{p} = -\mathbf{q}$ is a price at which $\mathbf{t}+\bar{\mathbf{s}}$ is demanded in the positive auction, and $\bar{\mathbf{s}}$ is demanded in the negative auction (see Lemma 1).

The DC Algorithm 1 finds a *stationary point* for two convex functions $f : \operatorname{dom} f \to \mathbb{R}$ and $g : \operatorname{dom} g \to \mathbb{R}$ with $\operatorname{dom} \partial f \subseteq \operatorname{dom} \partial g$ and $\operatorname{dom} \partial g^* \subseteq \operatorname{dom} \partial f^*$ (Tao and An, 1997). Our functions f and g, defined above, satisfy these conditions: By Appendix Lemma 4, $\operatorname{dom} f = \operatorname{conv} \Delta_{W_-} \subseteq \operatorname{conv}\{\mathbf{s} \in \mathbb{Z}^n : \mathbf{t}+\mathbf{s} \in \Delta_{W_+}\} = \operatorname{dom} g$, $\operatorname{dom} g^* = \mathbb{R}^n = \operatorname{dom} f^*$, and by Lemma 1 the domains of the respective functions coincide with the domains of their subdifferentials.

Algorithm 1: A DC-algorithm

```
Input: Convex functions $f : \operatorname{dom} f \to \mathbb{R}$, $g : \operatorname{dom} g \to \mathbb{R}$ with $\operatorname{dom} \partial f \subseteq \operatorname{dom} \partial g$
       and $\operatorname{dom} \partial g^* \subseteq \operatorname{dom} \partial f^*$
Output: Stationary points $\bar{\mathbf{s}} \in \mathbb{R}^n$ of $f - g$ and $\bar{\mathbf{q}}$ of $g^* - f^*$
Choose an initial $\mathbf{q}^0 \in \mathbb{R}^n$
for $k = 0, 1, \ldots$ do
    Choose $\mathbf{s}^k \in \partial f^*(\mathbf{q}^k)$
    Choose $\mathbf{q}^{k+1} \in \partial g(\mathbf{s}^k)$
    if $g^*(\mathbf{q}^{k+1}) - f^*(\mathbf{q}^{k+1}) = g^*(\mathbf{q}^k) - f^*(\mathbf{q}^k)$ then
        return $(\mathbf{s}^k, \mathbf{q}^k)$
    end if
end for
```

However, $\bar{\mathbf{s}}$ being a stationary point for f and g is not a *sufficient* condition for $\bar{\mathbf{s}}$ to globally minimize $f - g$. So we check whether a corresponding $\mathbf{p}$ is a local – and hence global – minimizer of the Lyapunov function L. If it is, then it is indeed an equilibrium price. If not, we go one step in the direction of steepest descent of the Lyapunov function and then restart the DC-algorithm. This is Algorithm 2 (where lines 1-8 are exactly Algorithm 1 with expressed in their economic interpretation; see Lemma 1 for more details).

Algorithm 2: DC auction algorithm

```
Input: Valid set $\mathcal{B}$ of SSPMA bids
Output: Equilibrium price $\mathbf{p}$ and supplementary bundle $\bar{\mathbf{s}}$
 1: Choose an initial price $\mathbf{p}^0$
 2: for $k = 0, 1, \ldots$ do
 3:     Choose a bundle $\mathbf{s}^k$ demanded at price $\mathbf{p}^k$ in the negative-bids auction
 4:     Choose an integral price vector $\mathbf{p}^{k+1}$ at which $\mathbf{t} + \mathbf{s}^k$ is demanded in the
        positive-bids auction
 5:     if $L(\mathbf{p}^{k+1}) = L(\mathbf{p}^k)$ then
 6:         return $(\mathbf{s}^k, \mathbf{p}^k)$
 7:     end if
 8: end for
 9: if there exists $\mathbf{e} \in \pm\{0,1\}^n$ with $L(\mathbf{p}^k + \mathbf{e}) < L(\mathbf{p}^k)$ then
10:     Restart the algorithm with $\mathbf{p}^0 := \mathbf{p}^k + \mathbf{e}$
11: end if
```

The value of $L(\mathbf{p}^k)$ decreases by at least one in every iteration 2-8 of the algorithm until the termination criterion in Step 5 is satisfied (we refer to Appendix A.5 for details). Whenever the algorithm is restarted in Step 10, L also decreases by at least one. Since there exists a minimizer for L, the algorithm terminates:

Theorem 3 *Algorithm 2 always terminates in a Walrasian equilibrium price.*

Algorithm 2 does not specify how to choose bundles $\mathbf{s}^k$ and prices $\mathbf{p}^{k+1}$. Determining bundles s^k is particularly simple when valuations are expressed in the SSPMA - we just allocate each bid with utility maximizing goods. For finding prices $\mathbf{p}^{k+1}$, an instance of (LP) must be solved. We use a min-cost flow solver to do so. Appendix A.6 explains our implementation in more detail.

Obtaining sharp worst-case bounds for Algorithm 2 is challenging due to the very generic nature of the DC-Algorithm 1. Note that the class of functions representable as a difference of convex functions is very large - for example, it contains all functions with continuous second derivative (Horst and Thoai, 1999). Also recall that Kobayashi (2015) shows that minimizing the difference of two general $M^\natural$-convex functions is NP-hard. Intuitively, we expect Algorithm 2 to perform particularly well when the number of negative bids is small. For example, when there are no negative bids at all, the algorithm boils down to solving the min-cost flow problem (LP). For the general case, we provide the following simple bound for Algorithm 2 by the number of negative bids.

First, observe that we may implement Step 3 to choose a bundle $\mathbf{s}^k$ which is uniquely demanded at some price–and indeed we do so in our practical implementation, because the vertices of demand sets $D_{|\mathcal{B}_-|}(\mathbf{p}^k)$ have this property.

We also assume that prices in Step 4 are chosen deterministically – for the same bundle, the algorithm always returns the same price.

Second, observe that if $\mathbf{s}^{k+1} = \mathbf{s}^k$, then the chosen prices in Step 4 are also equal, so the termination criterion 5 is satisfied. After a possible restart, the algorithm also can never reach this bundle again – this would contradict the strict monotonicity properties as we explain in the Appendix (Lemma 6). So in the worst case, after each restart of the algorithm, we directly choose bundles $\mathbf{s}^0 = \mathbf{s}^1$ in the first two iterations which immediately causes another restart. It follows that every possible bundle uniquely demanded in the negative auction is chosen at most twice in Step 3 of the algorithm.[30] If there is only one single negative bid, these are exactly $n+1$ bundles, and so the number of iterations, by which we mean the total number of iterations through the loop from Step 2 to Step 8, of Algorithm 2 is in $\mathcal{O}(n)$. Note that after each restart, we iterate at least once through the for loop, so the number of restarts is also in $\mathcal{O}(n)$. More generally, Proposition 3 shows that $\binom{n+|\mathcal{B}_-|}{n}$ bundles are demanded in total in the negative auction if the weights of all negative bids are equal to one. Since the number of uniquely demanded bundles does not change if we increase weights, $\binom{n+|\mathcal{B}_-|}{n}$ bounds the number of uniquely demanded bundles in general negative auctions. This therefore provides an upper bound on the number of bundles demanded uniquely in this auction, so on the number of iterations of Algorithm 2.

Proposition 5 *Algorithm 2 requires at most $\mathcal{O}\left(\binom{n+|\mathcal{B}_-|}{n}\right)$ iterations for solving the equilibrium pricing problem.*

This analysis gives a rather pessimistic worst-case bound for the algorithm, but it suggests that the algorithm performs particularly well with a low number of negative bids. In fact, in our experimental evaluation, we find that the DC algorithm is even faster than steepest descent in these environments.

When there are only positive bids, Algorithm 2 boils down to solving a single linear program, which can be formulated as a min-cost flow problem on a graph with $|V| = \mathcal{O}(n+|\mathcal{B}|)$ vertices and $|E| = \mathcal{O}(n|\mathcal{B}|)$ edges (see Appendix A.6). The enhanced capacity scaling algorithm (Ahuja et al., 1993) finds an optimal integral solution in

$$\mathcal{O}(|E|\log|V|(|E| + |V|\log|V|))$$

iterations, which implies that the pricing problem can be solved in time

$$\mathcal{O}(n^2|\mathcal{B}|^2\log^2(n+|\mathcal{B}|)) = \tilde{\mathcal{O}}(n^2|\mathcal{B}|^2)$$

this way. On the other hand, Baldwin, Goldberg, Klemperer, and Lock (2019) provide the worst-case bound $\mathcal{O}(n^2|\mathcal{B}|^2\log M + n|\mathcal{B}|T(n))$ for the steepest descent algorithm, where $M = \max_{\mathbf{b}\in\mathcal{B}}\|\mathbf{b}\|_\infty$ is the maximal price of a bid vector

[30] Moreover, if the same bundle is chosen twice, it is unnecessary to repeat step 4 – the most computationally costly part of the algorithm – so checking for $\mathbf{s}^{k+1} = \mathbf{s}^k$ provides a practical runtime improvement, although it does not alter the complexity class.

and $T(n)$ is the complexity of minimizing an n-dimensional submodular function. Note that the total asymptotic runtime of the network flow formulation coincides with the first summand $n^2|\mathcal{B}|^2 \log M$ of the steepest descent formulation up to a logarithmic factor. However, we may expect the second summand $n|\mathcal{B}|T(n)$ to dominate the runtime. To the best of our knowledge, the best known weakly polynomial worst case bound for minimizing a general integral submodular function is $\mathcal{O}(n^2 \log nU \cdot EO + n^3 \log^{\mathcal{O}(1)} nU)$, where U is an upper bound for the maximal value of the submodular function (Chakrabarty et al., 2017; Lee et al., 2015). Thus, the worst-case bound for $n|\mathcal{B}|T(n)$ cannot be better than $\mathcal{O}(n^4|\mathcal{B}|\log(nU))$ using known methods. Hence, in particular when the number of goods increases, we may expect the min-cost flow formulation to outperform the steepest descent formulation in the absence of negative bids.[31]

Paes Leme and Wong (2020) present a polynomial time algorithm for computing competitive equilibrium prices for bidders with general preferences and provide a specialization of their algorithm for gross substitutes valuations, which is the fastest algorithm for this setting currently known. They provide the worst-case bound

$$\begin{aligned} mn \cdot T_V + \mathcal{O}(mn \log m + n^3 \log(mnM) + n^3 \log^{\mathcal{O}(1)}(mnM)) = \\ = \tilde{\mathcal{O}}(mn \cdot T_V + n^3) \end{aligned}$$

for the runtime of their algorithm, where n is the number of goods, m is the number of bids, M is the maximum value a bid has for a bundle and T_V is the runtime of the value oracle. So the worst-case runtime of Paes Leme and Wong (2020)'s algorithm grows linearly in the number of bidders, while the worst-case runtime of our algorithm (when all its bids are positive) grows quadratically.[32] However, our algorithm performs much better than Paes Leme and Wong (2020)'s as we increase the number of units available without changing the number of goods (when all the SSPMA bids are positive), because its worst case is unaffected, whilst Paes Leme and Wong (2020)'s worst case is cubic in the number of units because they must treat each additional unit as an additional good.[33] Moreover, even when there are only small numbers of units available of each distinct good, we expect the use cases for the two algorithms to be different: in a setting where fast direct access to bidders' valuations is

[31] Baldwin, Goldberg, Klemperer, and Lock (2019)'s algorithm's worst case also depends on M, while our algorithm's does not, so our algorithm is more robust to increases in the precision with which valuations can be expressed (e.g, expressing valuations in cents rather than dollars multiplies M by 100).

[32] We assume that the runtime T_V of a value query is constant, and that the total number of SSPMA bids grows linearly in the number of agents.

[33] The worst-case bounds suggest our algorithm also performs better as we increase the number of goods, but this comparison is less clear. The analysis of our algorithm is for bidders with strong substitutes preferences expressed via positive SSPMA bids, and bidders may submit more SSPMA bids in markets with more goods. Although a bid in Paes Leme and Wong (2020)'s algorithm describes the entire valuation function of a bidder with general gross substitutes preferences, the runtime T_V of a value query may depend on the number of goods. Ignoring both these effects, our algorithm's worst case depends quadratically on the number of goods, while Paes Leme and Wong (2020)'s has cubic dependence.

possible, we expect applying Paes Leme and Wong (2020)'s algorithm would be preferable to first computing the corresponding SSPMA bids for every bidder and then applying our algorithm. On the other hand, when preferences are provided by bidders in the SSPMA bid language, we expect it is faster to use our algorithm directly, rather than translating the SSPMA bids into value oracles first and then using Paes Leme and Wong (2020).

4.2 Experimental Evaluation

We implemented both the DC auction algorithm and a steepest descent algorithm based on the Lyapunov function. The Lyapunov approach and the restart step in the DC algorithm require the minimization of a submodular function. As in Baldwin, Goldberg, Klemperer, and Lock (2019), we use the Fujishige-Wolfe algorithm (Chakrabarty et al., 2014), which in practice often outperforms other submodular minimization algorithms.

In our experimental evaluation we solved problems with 10-50 goods, 1020/1200/1500/3020/3500 positive and 20/200/500 negative bids. We drew on a specialization of the algorithm by Baldwin et al. (2016) to randomly generate valid groups of bids, each group consisting of 3 positive and 1 negative bids. Algorithm 4 in Appendix A.7 describes this procedure, and Table 2 in Appendix A.8 gives our results.

The DC algorithm appears to be faster if there are not too many negative bids (less than 200, in our experiments). Table 1 shows a selection of our results for the case of 20 negative bids. So if the number of negative bids is small, which we consider the most likely scenario (see section 2.3), our DC algorithm is a particularly good choice.

#pos. bids	#neg. bids	#goods	time DC (ms)	time SD (ms)
1020	20	10	31	394
1020	20	30	105	410
1020	20	50	206	665
3020	20	10	115	1152
3020	20	30	445	1278
3020	20	50	970	1366

Table 1 Runtimes of the DC and the steepest descent (SD)-algorithm for instances where the number of negative bids is low.

However, the main conclusion from Table 2 is that both algorithms are very fast, solving even the largest problems in our experiments in less than 3 seconds. Experiments with up to 50 goods and 10,000 bids can also be solved in a few seconds only.[34]

[34] Obviously our results are sensitive to the details of the implementations. In particular, in a first, textbook-style implementation, the steepest descent algorithm was much slower beyond 50 goods and 4000 bids. However, an additional pre-processing step led to significant

5 Conclusion

Strong substitutes valuations are of central importance for both theory and practical applications. We have developed a new algorithm for computing competitive equilibrium prices when agents' preferences are expressed using the Strong Substitutes Product-Mix Auction bidding language, a compact language that permits the expression of all strong-substitutes valuations (and no other valuations). By contrast with a previous approach of using a standard steepest-descent algorithm that tests candidate solutions in turn, we began from the economics of the problem. We used the fact that the shadow prices of two separate linear programs that maximize value for "positive" and "negative" bids, respectively, must be equal, and proved that our model formulation is dual to the Lyapunov function. We also used the bidding language to provide new insight into the geometric structure of strong substitutes valuations.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice hall (1993)

An, L.T.H., Tao, P.D.: The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. Annals of Operations Research **133**(1), 23–46 (Jan 2005), ISSN 1572-9338, doi:10.1007/s10479-004-5022-1, URL `https://doi.org/10.1007/s10479-004-5022-1`

Ausubel, L., Milgrom, P.: The lovely but lonely Vickrey auction. In: Cramton, P., Shoham, Y., Steinberg, R. (eds.) Combinatorial Auctions, chap. 1, pp. 17–40, MIT Press, Cambridge, MA (2006)

Ausubel, L.M.: An efficient dynamic auction for heterogeneous commodities. The American Economic Review pp. 602–629 (2006)

Baldwin, E., Goldberg, P., Klemperer, P.: Valid combinations of bids. Tech. rep., Oxford University (2016)

Baldwin, E., Goldberg, P.W., Klemperer, P., Lock, E.: Solving strong-substitutes product-mix auctions. arXiv preprint arXiv:1909.07313 (2019)

Baldwin, E., Klemperer, P.: Proof that any strong substitutes preferences can be represented by the strong substitutes product-mix auction bidding language. Tech. rep., (day 2 slides on Tropical intersections and equilibrium, Baldwin, E., Goldberg, P. and Klemperer, P.) Hausdorff School on Tropical Geometry and Economics (2016), URL `http://people.math.gatech.edu/~jyu67/HCM/Baldwin2.pdf`

Baldwin, E., Klemperer, P.: Understanding preferences: "demand types," and the existence of equilibrium with indivisibilities. Econometrica **87**(3), 867–932 (2019)

improvements in the steepest descent algorithm, and we report the results for this improved steepest descent algorithm. With this improvement in the steepest descent algorithm, the differences between the algorithms seem likely to be small in most applications.

Baldwin, E., Klemperer, P.: Proof that the strong substitutes product-mix auction bidding language can represent any strong substitutes preferences. (2021), available from `http://elizabeth-baldwin.me.uk/papers/strongsubsproof.pdf` and `http://www.nuff.ox.ac.uk/users/klemperer/strongsubsproof.pdf`

Beck, M., Robins, S.: Computing the Continuous Discretely - Integer-point Enumeration in Polyhedra. Springer Science & Business Media, Berlin Heidelberg (2007), ISBN 978-0-387-29139-0

Bichler, M., Fichtl, M., Schwarz, G.: Walrasian equilibria from an optimization perspective: A guide to the literature. Naval Research Logistics (NRL) **n/a**(n/a) (2020), doi:https://doi.org/10.1002/nav.21963

Bichler, M., Shabalin, P., Ziegler, G.: Efficiency with linear prices? a game-theoretical and computational analysis of the combinatorial clock auction. Information Systems Research **24**(2), 394–417 (2013)

Bikhchandani, S., Mamer, J.W.: Competitive equilibrium in an exchange economy with indivisibilities. Journal of economic theory **74**(2), 385–413 (1997)

Bikhchandani, S., Ostroy, J.M.: The package assignment model. Journal of Economic theory **107**(2), 377–406 (2002)

Chakrabarty, D., Jain, P., Kothari, P.: Provable submodular minimization using Wolfe's algorithm. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 802–809, Curran Associates, Inc. (2014)

Chakrabarty, D., Lee, Y.T., Sidford, A., Wong, S.C.w.: Subquadratic submodular function minimization. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pp. 1220–1231, STOC 2017, Association for Computing Machinery, New York, NY, USA (2017), ISBN 9781450345286, doi:10.1145/3055399.3055419, URL `https://doi.org/10.1145/3055399.3055419`

Danilov, V., Koshevoy, G., Murota, K.: Discrete convexity and equilibria in economies with indivisible goods and money. Mathematical Social Sciences **41**(3), 251–273 (2001)

Fichtl, M.: On the expressiveness of assignment messages. Economics Letters **208**, 110051 (2021), ISSN 0165-1765, doi:https://doi.org/10.1016/j.econlet.2021.110051, URL `https://www.sciencedirect.com/science/article/pii/S0165176521003281`

Goetzendorff, A., Bichler, M., Shabalin, P., Day, R.W.: Compact bid languages and core pricing in large multi-item auctions. Management Science **61**(7), 1684–1703 (2015)

Goldberg, P.W., Lock, E., Marmolejo-Cossío, F.J.: Learning strong substitutes demand via queries. In: Chen, X., Gravin, N., Hoefer, M., Mehta, R. (eds.) Web and Internet Economics - 16th International Conference, WINE, Lecture Notes in Computer Science, vol. 12495, pp. 401–415, Springer (2020), doi:10.1007/978-3-030-64946-3_28

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

Hatfield, J.W., Milgrom, P.R.: Matching with contracts. American Economic Review **95**(4), 913–935 (September 2005)

Hauser, A.: The future of repo: 'too much' or 'too little'? (June 2013), URL `https://www.bankofengland.co.uk/-/media/boe/files/speech/2013/the-future-of-repo-too-much-or-too-little`, speech at the ICMA Conference on the Future of the Repo Market

Horst, R., Thoai, N.V.: Dc programming: overview. Journal of Optimization Theory and Applications **103**(1), 1–43 (1999)

Kelso, A.S., Crawford, V.P.: Job matching, coalition formation, and gross substitutes. Econometrica **50**, 1483–1504 (1982)

Klemperer, P.: A new auction for substitutes: Central bank liquidity auctions, the U.S. TARP, and variable product-mix auctions. Tech. Rep. 1, Oxford University (2008)

Klemperer, P.: The product-mix auction: A new auction design for differentiated goods. Journal of the European Economic Association **8**(2-3), 526–536 (2010)

Klemperer, P.: Product-mix auctions. Working Paper 2018-W07, Nuffield College (2018)

Kobayashi, Y.: The complexity of minimizing the difference of two m-convex set functions. Operations Research Letters **43**, 573–574 (11 2015), doi:10.1016/j.orl.2015.08.011

Lee, Y.T., Sidford, A., Wong, S.: A faster cutting plane method and its implications for combinatorial and convex optimization. In: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, pp. 1049–1065 (10 2015), doi:10.1109/FOCS.2015.68

Lin, B., Tran, N.M.: Linear and rational factorization of tropical polynomials. arXiv preprint arXiv:1707.03332 (2017)

Maehara, T., Marumo, N., Murota, K.: Continuous relaxation for discrete dc programming. Mathematical Programming **169**(1), 199–219 (2018)

Maehara, T., Murota, K.: A framework of discrete dc programming by discrete convex analysis. Mathematical Programming **152**(1), 435–466 (2015)

Milgrom, P.: Assignment messages and exchanges. American Economic Journal: Microeconomics **1**(2), 95–113 (2009)

Milgrom, P., Strulovici, B.: Substitute goods, auctions, and equilibrium. Journal of Economic theory **144**(1), 212–247 (2009)

Murota, K.: Discrete convex analysis, vol. 10. SIAM (2003)

Murota, K.: Discrete convex analysis: A tool for economics and game theory. Journal of Mechanism and Institution Design **1**(1), 151–273 (2016)

Murota, K., Tamura, A.: New characterizations of M-convex functions and their applications to economic equilibrium models with indivisibilities. Discrete Applied Mathematics **131**(2), 495–512 (2003)

Nisan, N., Segal, I.: The communication requirements of efficient allocations and supporting prices. Journal of Economic Theory **129**(1), 192–224 (2006)

Ostrovsky, M., Paes Leme, R.: Gross substitutes and endowed assignment valuations. Theoretical Economics **10**(3), 853–865 (2015)

Paes Leme, R.: Gross substitutability: An algorithmic survey. Games and Economic Behavior **106**, 294–316 (2017)

Paes Leme, R., Wong, S.C.w.: Computing Walrasian equilibria: fast algorithms and structural properties. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 632–651 (2017)

Paes Leme, R., Wong, S.C.w.: Computing walrasian equilibria: Fast algorithms and structural properties. Mathematical Programming **179**(1), 343–384 (2020)

Rockafellar, R.T.: Convex analysis. 28, Princeton university press (1970)

Schneider, R.: Convex Bodies: The Brunn-Minkowski Theory. Cambridge University Press, Cambridge, 2 edn. (2013)

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

Tao, P.D., An, L.T.H.: Convex analysis approach to dc programming: Theory, algorithms and applications. Acta Mathematica Vietnamica **22**(1), 289–355 (1997)

Toland, J.F.: A duality principle for non-convex optimisation and the calculus of variations. Archive for Rational Mechanics and Analysis **71**(1), 41–61 (1979)

Tran, N.M.: The finite matroid-based valuation conjecture is false (2019)

A Appendix: Additional Proofs

A.1 Proof of Proposition 1

We need the following simple Lemmas.

Lemma 2 *[See, e.g. Schneider (2013) Lemma 3.1.11] Let $A, B \subseteq \mathbb{R}^n$ be convex. Then $(A + B) - B = A$.*

Lemma 3 *Suppose $(\operatorname{conv} A) \cap \mathbb{Z}^n = A$ and $(\operatorname{conv} B) \cap \mathbb{Z}^n = B$. Then $(\operatorname{conv} A - \operatorname{conv} B) \cap \mathbb{Z}^n = A - B$.*

Proof. If $\mathbf{x} \in (\operatorname{conv} A - \operatorname{conv} B) \cap \mathbb{Z}^n$ then $\mathbf{x} \in \mathbb{Z}^n$ and $\mathbf{x} + \operatorname{conv} B \subseteq \operatorname{conv} A$, so $\mathbf{x} + (\operatorname{conv} B) \cap \mathbb{Z}^n \subseteq (\operatorname{conv} A) \cap \mathbb{Z}^n$, and therefore $\mathbf{x} + B \subseteq A$. So $\mathbf{x} \in A - B$. Conversely, if $\mathbf{x} \in A - B$ then $\mathbf{x} \in Z^n$, and $\mathbf{x} + B \subseteq A$ implies $\operatorname{conv}(\mathbf{x} + B) = \mathbf{x} + \operatorname{conv} B \subseteq \operatorname{conv} A$. □

Proof of Proposition 1. By the strong-substitutes property, the sets $D_{\mathcal{B}_+}(\mathbf{p})$ and $D_{|\mathcal{B}_-|}(\mathbf{p})$ are equal to the set of integer points of their respective convex hull, as by definition is $D_{\mathcal{B}}(\mathbf{p})$. So if we can show that $\operatorname{conv} D_{\mathcal{B}}(\mathbf{p}) + \operatorname{conv} D_{|\mathcal{B}_-|}(\mathbf{p}) = \operatorname{conv} D_{\mathcal{B}_+}(\mathbf{p})$, this implies by Lemma 2 that $\operatorname{conv} D_{\mathcal{B}}(\mathbf{p}) = \operatorname{conv} D_{\mathcal{B}_+}(\mathbf{p}) - \operatorname{conv} D_{|\mathcal{B}_-|}(\mathbf{p})$ and by Lemma 3 consequently that $D_{\mathcal{B}}(\mathbf{p}) = D_{\mathcal{B}_+}(\mathbf{p}) - D_{|\mathcal{B}_-|}(\mathbf{p})$.

But, as $\mathcal{B}$ is a valid set of bids, we know by Baldwin, Goldberg, Klemperer, and Lock (2019) Theorem 2.3 that $u_{\mathcal{B}}$ is the indirect utility of a strong substitutes valuation v such that $D_v(\mathbf{p}) = D_{\mathcal{B}}(\mathbf{p})$ for all $\mathbf{p} \in \mathbb{R}^n$. It follows that each vertex $\mathbf{x}$ of $D_{\mathcal{B}}(\mathbf{p})$ is the unique element of $D_{\mathcal{B}}(\mathbf{q})$ for a price $\mathbf{q}$ close to $\mathbf{p}$ and such that $\mathbf{x}$ minimizes $(\mathbf{q} - \mathbf{p}) \cdot \mathbf{x}'$ for $\mathbf{x}' \in D_{\mathcal{B}}(\mathbf{p})$. But similarly the minimizers of $(\mathbf{q} - \mathbf{p}) \cdot \mathbf{x}'$ for $\mathbf{x}' \in D_{\mathcal{B}_+}(\mathbf{p})$ and $\mathbf{x}' \in D_{|\mathcal{B}_-|}(\mathbf{p})$ are, respectively, the unique elements of $D_{\mathcal{B}_+}(\mathbf{p})$ and $D_{|\mathcal{B}_-|}(\mathbf{p})$. By definition, we have

$D_{\mathcal{B}}(\mathbf{q}) = D_{\mathcal{B}_+}(\mathbf{q}) - D_{|\mathcal{B}_-|}(\mathbf{q})$, so $D_{\mathcal{B}}(\mathbf{q}) + D_{|\mathcal{B}_-|}(\mathbf{q}) = D_{\mathcal{B}_+}(\mathbf{q})$. As this holds for all extreme points of $D_{\mathcal{B}}(\mathbf{p})$, it follows that $\operatorname{conv} D_{\mathcal{B}}(\mathbf{p}) + \operatorname{conv} D_{|\mathcal{B}_-|}(\mathbf{p}) = \operatorname{conv} D_{\mathcal{B}_+}(\mathbf{p})$, as required. □

A.2 The valuation v_r from Ostrovsky and Paes Leme (2015)

We now explain the construction of v_r from Ostrovsky and Paes Leme (2015). Let $G = (V, E)$ be an undirected graph with 4 vertices and 6 edges $E = \{1, \dots, 6\}$, such that every vertex is connected to every other vertex by an edge (see Figure 5). A subset H of E is called *independent* if it contains no cycles. For any $H \subseteq E$, the *rank* of H is the maximal cardinality of an independent subset contained in H:

$$\operatorname{rank}(H) = \max\left\{|H'| : H' \subseteq H \text{ is independent}\right\}.$$

The rank function induces the valuation $v_r : \{0,1\}^6 \to \mathbb{Z}$ given by $v_r(\mathbf{x}) = \operatorname{rank}(\{i : x_i = 1\})$. As Ostrovsky and Paes Leme (2015) show, v_r is strong substitutes. However, it does not satisfy the property of *strong exchangeability* which, as Ostrovsky and Paes Leme (2015) show, is a characteristic of every endowed assignment valuation. Consequently, it is not possible to express v_r by endowed assignment messages. We demonstrate, however, that it can be expressed using the SSPMA. Note that valuations induced by SSPMA bids are always defined on a scaled simplex Δ_W for some total weight $W \in \mathbb{Z}_{\geq 0}$. We thus naturally extend v_r to $\Delta_6 \supseteq \{0,1\}^6$ by assuming free disposal: $v_r(\mathbf{x}) = \operatorname{rank}(\{i : x_i \geq 1\})$.

Proof of Proposition 2. Given $H \subseteq E$, we write $\mathbf{b}^H := \sum_{i \in H} \mathbf{e}_i$. We make the following bids:

0. Place a bid $\mathbf{b}^{\emptyset}$ with $w(\mathbf{b}^{\emptyset}) = 3$
1. For all $H \subseteq E$ with $|H| = 3$ and H^c is a cycle in G, make a bid $\mathbf{b}^H$ with $w(\mathbf{b}^H) = 1$.
2. For all $H \subseteq E$ constituting a cycle of length 4, make a bid $\mathbf{b}^H$ with $w(\mathbf{b}^H) = 1$.
3. For all $H \subseteq E$ with $|H| = 5$ make a bid $\mathbf{b}^H$ with $w(\mathbf{b}^H) = -1$.
4. Make a bid $\mathbf{b}^E$ with $w(\mathbf{b}^E) = 2$.

Denote by $v_r(\mathbf{x}) = \operatorname{rank}(\{i : x_i \geq 1\})$ for $\mathbf{x} \in \Delta_6$ the valuation induced by the rank function, and by $v_{\mathcal{B}}(\mathbf{x})$ the valuation induced by the above bids. Our goal is to show $v_{\mathcal{B}} = v_r$. Note that bid 0 only ensures that the domains of v_r and $v_{\mathcal{B}}$ are equal, and does not "contribute" to the valuations apart from this. So let us check that indeed $\operatorname{dom} v_{\mathcal{B}} = \Delta_6$. There is 1 bid of type 0, 4 bids of type 1, 3 bids of type 2, and 1 bid of type 4. So summing up the weights of these bids gives $W_+ = 12$. On the other hand, there are 6 bids of type 3, so $W_- = 6$, and consequently $\operatorname{dom} v_{\mathcal{B}} = \Delta_{12-6} = \Delta_6$.

We have $u_r(\mathbf{p}) = \max_{\mathbf{x} \in \Delta_6} v_r(\mathbf{x}) - \langle \mathbf{p}, \mathbf{x} \rangle$ and $u_{\mathcal{B}}$ is defined by Equation (3). Recall from Section 2.1 that, for $i \in \{r, \mathcal{B}\}$, we have $v_i(\mathbf{x}) = \min_{\mathbf{p} \in \mathbb{R}^6} u_i(\mathbf{p}) + \langle \mathbf{p}, \mathbf{x} \rangle$, where one can check that $\mathbf{p} \mapsto u_i(\mathbf{p}) + \langle \mathbf{p}, \mathbf{x} \rangle$ always possesses a non-negative minimizer $\mathbf{p}$ for $\mathbf{x} \in \Delta_6$.

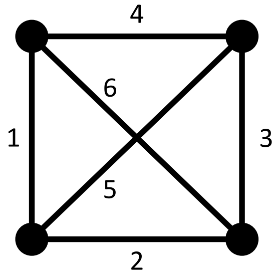

Fig. 5 Graph used to construct the valuation v_r from Ostrovsky and Paes Leme (2015).

So in order to prove Proposition 2, it suffices to show that $u_r(\mathbf{p}) = u_{\mathcal{B}}(\mathbf{p})$ for all $\mathbf{p} \in \mathbb{R}^6_{\geq 0}$. By L$^\natural$-convexity of u_r and $u_{\mathcal{B}}$ (Murota, 2003), both are determined uniquely on $\mathbb{R}^6_{\geq 0}$ by the values $u_r(\mathbf{p})$ and, respectively, $u_{\mathcal{B}}(\mathbf{p})$ for $\mathbf{p} \in \mathbb{Z}^6_{\geq 0}$. Moreover, given $\mathbf{p} \in \mathbb{Z}^6_{\geq 0}$, define $\tilde{\mathbf{p}}$ by $\tilde{p}_i = p_i$ if $p_i \leq 1$ and $\tilde{p}_i = 1$, otherwise. Since the marginal value of any good is at most 1 for v_r, and no bid in $\mathcal{B}$ has any value greater than 1, allocating a good i with $\tilde{p}_i = 1$ can never increase utilities, so we have $u_r(\mathbf{p}) = u_r(\tilde{\mathbf{p}})$ and $u_{\mathcal{B}}(\mathbf{p}) = u_{\mathcal{B}}(\tilde{\mathbf{p}})$. So our problem reduces to showing that $u_r(\mathbf{p}) = u_{\mathcal{B}}(\mathbf{p})$ for all $\mathbf{p} \in \{0,1\}^6$. For $H \subseteq \{1,\dots,6\}$, denote by $\mathbf{p}^H \in \{0,1\}^6$ the price vector with $p_i^H = 1$ if and only if $i \in H$. We will show that $u_r(\mathbf{p}^H) = u_{\mathcal{B}}(\mathbf{p}^H)$ for all $H \subseteq \{1,\dots,6\}$.

We claim that $u_r(\mathbf{p}^H) = \operatorname{rank}(H^c)$. To see this, let $\mathbf{x}$ be a bundle with $u_r(\mathbf{p}^H) = v_r(\mathbf{x}) - \langle \mathbf{p}^H, \mathbf{x} \rangle = v_r(\mathbf{x}) - \sum_{i \in H} \mathbf{x}_i$. Let $P = \{i : x_i \geq 1\}$. Then

$$\begin{aligned} u_r(\mathbf{p}^H) = \operatorname{rank}(P) - |P \cap H| &\leq \operatorname{rank}(P \cap H^c) + \operatorname{rank}(P \cap H) - |P \cap H| \\ &\leq \operatorname{rank}(P \cap H^c) + |P \cap H| - |P \cap H| \\ &\leq \operatorname{rank}(H^c) = v_r\left(\sum_{i \in H^c} \mathbf{e}_i\right) - 0 \leq u_r(\mathbf{p}^H) \end{aligned}$$

by properties of matroid rank functions. Consequently, equality must hold everywhere, so $u_r(\mathbf{p}^H) = \operatorname{rank}(H^c)$.

Regarding the indirect utility of our bids, we observe that at prices $\mathbf{p}^H$, the bid $\mathbf{b}^{\tilde{H}}$ generates a utility of $w(\mathbf{b}^{\tilde{H}})$ if and only if $\tilde{H} \cap H^c \neq \emptyset$, i.e., if and only if $\mathbf{b}^{\tilde{H}}$ has positive value for at least one good not in H. Otherwise it generates utility 0.

We now consider all subsets $H \subseteq \{1,\dots,6\}$ and show that in each case, $u_r(\mathbf{p}^H) = u_{\mathcal{B}}(\mathbf{p}^H)$.

First, for price vectors $\mathbf{p}^H$ with $|H| < 3$, all bids are accepted, since every placed bid has positive values for at least 3 goods. There are 4 bids of type 1, 3 bids of type 2, 6 bids of type 3 and 1 bid of type 4. In total, we get

$$u_b(\mathbf{p}^H) = 4 \cdot 1 + 3 \cdot 1 + 6 \cdot (-1) + 1 \cdot 2 = 3.$$

On the other hand, one can see from Figure 5 that every subset containing at least 4 edges contains a cycle free subset of cardinality 3, and there is no cycle free subset with more than 3 elements. Consequently, $u_r(\mathbf{p}^H) = \operatorname{rank}(H^c) = 3$.

Now consider $\mathbf{p}^H$ with $|H| = 3$. Obviously, all bids on more than 3 edges get accepted. A bid $\mathbf{b}^{\tilde{H}}$ with $\tilde{H} = 3$ is rejected, if and only if $H = \tilde{H}$. In this case, H^c is a cycle of length 3, so $u_r(\mathbf{p}^H) = \operatorname{rank}(H^c) = 2$. We then also clearly have $u_{\mathcal{B}}(\mathbf{p}^H) = 2$, since exactly one bid is rejected, and all others are accepted.

On the other hand, if $|H| = 3$ and no bid is rejected, H^c is cycle free, so $u_r(\mathbf{p}^H) = \operatorname{rank}(H^c) = 3 = u_{\mathcal{B}}(\mathbf{p}^H)$.

Next, suppose $|H| = 4$, so $u_r(\mathbf{p}^H) = \operatorname{rank}(H^c) = 2$, because 2 edges cannot form a cycle. Regarding the bids, if H is a cycle of length 4, one bid of type 2 is rejected. In this case, H^c consists of two non adjacent edges. Consequently, there is no $i \in H$ such that $\{i\} \cup H^c$ is a cycle. Equivalently, for no $\tilde{H} \subseteq H$ with $|\tilde{H}| = 3$ we have that $\tilde{H}^c$ is a cycle, so no bid of type 1 is rejected, and $u_{\mathcal{B}}(\mathbf{p}^H) = 2$.

If, otherwise, H has no cycle of length 4, H^c consists of two adjacent edges. Thus, there is a unique $e \in H$ with $\{i\} \cup H^c$ being a cycle, so a single bid of type 1 is rejected, which means that again $u_{\mathcal{B}}(\mathbf{p}^H) = 2$.

Concerning $|H| = 5$, since the graph is complete, we can assume by symmetry that $H = \{1,2,3,4,5\}$. Then the bids $\mathbf{b}^{\tilde{H}}$ with

$$\tilde{H} \in \{\{1,2,5\}, \{3,4,5\}, \{1,2,3,4\}, \{1,2,3,4,5\}\}$$

are rejected, and $u_{\mathcal{B}}(\mathbf{p}^H) = 2 \cdot 1 + 2 \cdot 1 + 5 \cdot (-1) + 1 \cdot 2 = 1 = \operatorname{rank}(H^c) = u_r(\mathbf{p}^H)$.

Finally, for $H = E$, all bids are rejected, so $u_{\mathcal{B}}(\mathbf{p}^H) = u_r(\mathbf{p}^H) = 0$.

We have shown that for all $p \in \{0,1\}^6$, $u_{\mathcal{B}}(\mathbf{p}) = u_r(\mathbf{p})$, which proves our statement. □

A.3 Proof of Proposition 3

Proof of Proposition 3. We will show that $D = \mathit{\Delta}_W = \mathbb{Z}^n \cap W\mathit{\Delta}$, where $\mathit{\Delta}$ is the standard simplex in dimension n, spanned by $\mathbf{0}$ and the standard unit vectors $\mathbf{e}_i$. Since $W\mathit{\Delta}$ contains exactly $\binom{n+W}{n}$ integer points (Beck and Robins, 2007, Theorem 2.2), the remaining results follow.

By the strong substitutes property, $D = (\operatorname{conv} D) \cap \mathbb{Z}^n$, so it suffices to show that $\operatorname{conv} D = W\mathit{\Delta}$. To that goal we note that if we set $p_i = -1$ and p_j very large for $j \neq i$, then $D(\mathbf{p}) = \{W\mathbf{e}_i\}$, since every bid $\mathbf{b}$ is allocated with $w(\mathbf{b})$ items of good i and the total weight of all bids is W. Also, for a very large price (in every coordinate) $\mathbf{p}$, we have $D_{\mathcal{B}}(\mathbf{p}) = \{\mathbf{0}\}$. Consequently, $\operatorname{conv} D \supseteq W\mathit{\Delta}$. To see the reverse inclusion, note that any demanded bundle cannot contain strictly more than W items, as some bid would have to be allocated with more than $w(\mathbf{b})$ items otherwise. The lower bounds come from the basic inequality $\binom{m}{k} \geq (m/k)^k$. □

A.4 Proof of Theorem 1

Proof of Lemma 1. The linear program (LP) of Section 3 is clearly defined for any $\mathbf{x} = \mathbf{t} \in \operatorname{conv} \mathit{\Delta}_W$, and we can use this to assign a real value to $\tilde{v}_{\mathcal{B}}(\mathbf{x})$ for $\mathbf{x} \in \operatorname{conv} \mathit{\Delta}_W$ and set $f = -\tilde{v}_{\mathcal{B}}$. Since f is a polyhedral convex function according to (Rockafellar, 1970, p. 172), its subdifferential is nonempty at every point of $\operatorname{dom} f$ (Rockafellar, 1970, Theorem 23.10), so $\operatorname{dom} \partial f = \operatorname{dom} f = \operatorname{conv} \mathit{\Delta}_W$. Let us consider the convex conjugate f^* of $f(\mathbf{x}) = -\tilde{v}_{\mathcal{B}}(\mathbf{x})$. By definition, $f^*(\mathbf{q}) = \max_{\mathbf{x} \in \operatorname{conv} \mathit{\Delta}_W} \langle \mathbf{q}, \mathbf{x} \rangle + \tilde{v}_{\mathcal{B}}(\mathbf{x})$, or in LP-form:

$$
\begin{aligned}
f^*(\mathbf{q}) = \max \sum_{\mathbf{b} \in \mathcal{B}} \sum_{i \in [n]} (b_i + q_i) y_{\mathbf{b}i} & \\
\text{s.t. } x_i = \sum_{\mathbf{b} \in \mathcal{B}} y_{\mathbf{b}i} & \qquad \forall i \in [n] \\
\sum_{i \in [n]} y_{\mathbf{b}i} \leq w(\mathbf{b}) & \qquad \forall \mathbf{b} \in \mathcal{B} \\
y_{\mathbf{b}i} \geq 0 & \qquad \forall \mathbf{b} \in \mathcal{B},\, i \in [n].
\end{aligned}
$$

Note that since the set of feasible solutions $\mathbf{x}$ is compact, $f^*(\mathbf{q})$ attains a finite value for all $\mathbf{q} \in \mathbb{R}^n$, so $\operatorname{dom} \partial f^* = \operatorname{dom} f^* = \mathbb{R}^n$, since f^* is also polyhedral convex. Let us now derive the expressions for ∂f and ∂f^*. To that goal, note that $\mathbf{x}$ maximizes the above linear program if and only if $\mathbf{x} \in \partial f^*(\mathbf{q})$, which is in turn equivalent to $\mathbf{q} \in \partial f(\mathbf{x})$ (Rockafellar, 1970, Theorem 23.5). It is not hard to see from Equations (1) and (2) that the variables $y_{\mathbf{b}i}$ constitute an optimal solution for the above linear program, if and only if for every fixed $\mathbf{b}$ the vector $(y_{\mathbf{b}i})_{i=1}^n$ lies in $\operatorname{conv} D_{\mathbf{b}}(-\mathbf{q})$, which can be seen to be equivalent to $\mathbf{x} \in \operatorname{conv} D_{\mathcal{B}}(-\mathbf{q})$ (recall that in the case of only positive bids, the aggregate demand set is just the Minkowski sum of the individual demand sets). It now directly follows that $\partial f^*(\mathbf{q}) = \operatorname{conv} D_{\mathcal{B}}(-\mathbf{q})$ and $\partial f(\mathbf{x}) = -\{\mathbf{p} : \mathbf{x} \in \operatorname{conv} D_{\mathcal{B}}(\mathbf{p})\}$.

□

Lemma 4 *Let $\mathcal{B}$ be a valid collection of bids. Let $\mathbf{t} \in \mathit{\Delta}_W$ and $\mathbf{s} \in \mathit{\Delta}_{W_-}$. Then $\mathbf{t} + \mathbf{s} \in \mathit{\Delta}_{W_+}$. Consequently, for the convex extensions f of $\mathbf{s} \mapsto -v_{|\mathcal{B}_-|}(s)$ and g of $\mathbf{s} \mapsto -v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s})$ we have that* $\operatorname{dom} f = \operatorname{conv} \mathit{\Delta}_{W_-} \subseteq \operatorname{conv}\{s \in \mathbb{Z}^n : \mathbf{t} + \mathbf{s} \in \mathit{\Delta}_{W_+}\} = \operatorname{dom} g$.

Proof. As $\mathbf{t} \in \mathit{\Delta}_W$, we have $\sum_{i=1}^n t_i \leq W$. Similarly, $\sum_{i=1}^n s_i \leq W_-$. Since $W = W_+ - W_-$ it follows that $\sum_{i=1}^n (t_i + s_i) \leq W_+$, so $\mathbf{t} + \mathbf{s} \in \mathit{\Delta}_{W_+}$. This directly implies the second part of the Lemma.

□

Proof of Theorem 1. Let f be the convex extension of $\mathbf{s} \mapsto -v_{|\mathcal{B}_-|}(\mathbf{s})$ and g the convex extension of $\mathbf{s} \mapsto -v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s})$. Then $\operatorname{dom} f = \operatorname{conv} \Delta_{W_-} \subseteq \operatorname{dom} g$ and $\operatorname{dom} g^* = \mathbb{R}^n = f^*$ by Lemmas 1 and 4. From Lemma 1 we know that $f^*(\mathbf{q}) = u_{|\mathcal{B}_-|}(-\mathbf{q})$. Similarly, $g^*(\mathbf{q}) = u_{\mathcal{B}_+}(-\mathbf{q}) - \langle \mathbf{q}, \mathbf{t}\rangle$. So we can apply Theorem 2 to $f - g$ and get

$$\min_{\mathbf{s}\in \operatorname{conv} \Delta_{W_-}} \tilde{v}_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - \tilde{v}_{|\mathcal{B}_-|}(\mathbf{s}) = \min_{\mathbf{q}\in\mathbb{R}^n} u_{\mathcal{B}_+}(-\mathbf{q}) - u_{|\mathcal{B}_-|}(-\mathbf{q}) - \langle \mathbf{q}, \mathbf{t}\rangle,$$

if any of the two problems has a solution. By substituting $\mathbf{p} = -\mathbf{q}$, we can rewrite the problem on the right as

$$\min_{\mathbf{p}\in\mathbb{R}^n} u_{\mathcal{B}_+}(\mathbf{p}) - u_{|\mathcal{B}_-|}(\mathbf{p}) + \langle \mathbf{p}, \mathbf{t}\rangle = \min_{\mathbf{p}\in\mathbb{R}^n} u_{\mathcal{B}}(\mathbf{p}) + \langle \mathbf{p}, \mathbf{t}\rangle.$$

The expression $u_{\mathcal{B}}(\mathbf{p}) + \langle \mathbf{p}, \mathbf{t}\rangle$ is exactly the Lyapunov function $L(\mathbf{p})$ introduced in Section 3. For strong-substitutes valuations, the Lyapunov function always attains a minimum, and the set of minimizers is equal to the set of equilibrium prices for the target $\mathbf{t}$ (Ausubel and Milgrom, 2006). Consequently, the problem $\min_{\mathbf{s}\in \operatorname{conv} \Delta_{W_-}} \tilde{v}_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - \tilde{v}_{|\mathcal{B}_-|}(\mathbf{s})$ also has a solution $\mathbf{s} \in \operatorname{conv} \Delta_{W_-}$, and the values of both minimization problems are equal. There exists at least one integral solution $\bar{s} \in \Delta_W$ to this problem: Let $\mathbf{p}$ be a minimizer of the Lyapunov function. By Theorem 2, each $\mathbf{s} \in \partial f^*(-\mathbf{p}) = \operatorname{conv} D_{\mathcal{B}}(\mathbf{p})$ minimizes $\tilde{v}_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - \tilde{v}_{|\mathcal{B}_-|}(\mathbf{s})$, so in particular each $\bar{\mathbf{s}} \in D_{\mathcal{B}}(\mathbf{p}) \neq \emptyset$ does so. Since the valuations $v_{\mathcal{B}_+}$ and $v_{|\mathcal{B}_-|}$ coincide on integral bundles with $\tilde{v}_{\mathcal{B}_+}$ and $\tilde{v}_{|\mathcal{B}_-|}$ by construction,

$$\min_{\mathbf{s}\in \Delta_{W_-}} v_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - v_{|\mathcal{B}_-|}(\mathbf{s}) = \min_{\mathbf{s}\in \operatorname{conv} \Delta_{W_-}} \tilde{v}_{\mathcal{B}_+}(\mathbf{t}+\mathbf{s}) - v_{|\mathcal{B}_-|}(\mathbf{s}) = \min_{\mathbf{p}\in\mathbb{R}^n} L(\mathbf{p}).$$

Finally, again by Theorem 2, if $\bar{s} \in \Delta_{W_-}$ is a minimizer, each $\mathbf{p}$ with $-\bar{\mathbf{p}} \in \partial g(\mathbf{t}+\mathbf{s}) = -\{\mathbf{p} \in \mathbb{R}^n : \mathbf{t} + \bar{s} \in \operatorname{conv} D_{\mathcal{B}_+}(\mathbf{p})\}$ minimizes L. In other words, each equilibrium price for $\mathbf{t}+\mathbf{s}$ for the positive auction is an equilibrium price for the complete auction as well. □

A.5 Proof of Theorem 3

We now prove Theorem 3 which states that our DC-algorithm always terminates in a global minimum. First, we collect some properties of the DC-algorithm.

Proposition 6 *Algorithm 1 has the following properties*

1. *The sequences $f(\mathbf{s}^k) - g(\mathbf{s}^k)$ and $g^*(\mathbf{q}^k) - f^*(\mathbf{q}^k)$ are decreasing. Furthermore, $f(\mathbf{s}^k) - g(\mathbf{s}^k) \leq g^*(\mathbf{q}^k) - f^*(\mathbf{q}^k)$ and $g^*(\mathbf{q}^{k+1}) - f^*(\mathbf{q}^{k+1}) \leq f(\mathbf{s}^k) - g(\mathbf{s}^k)$. The sequence $g^*(\mathbf{q}^k) - f^*(\mathbf{q}^k)$ is strictly decreasing until the termination criterion is met.*
2. *If the algorithm terminates with $(\mathbf{s}^k, \mathbf{q}^k)$, then $\mathbf{s}^k \in \partial f^*(\mathbf{q}^k) \cap \partial g^*(\mathbf{q}^k)$ and $\mathbf{q}^k \in \partial f(\mathbf{s}^k) \cap g(\mathbf{s}^k)$. Furthermore, $f(\mathbf{s}^k) - g(\mathbf{s}^k) = g^*(\mathbf{q}^k) - f^*(\mathbf{q}^k)$.*

Proof. A proof can be found in (Tao and An, 1997, Theorem 3). The sequence $g^*(\mathbf{q}^k) - f^*(\mathbf{q}^k)$ is strictly decreasing because of the algorithm terminates as soon as $g^*(\mathbf{q}^k) - f^*(\mathbf{q}^k) = g^*(\mathbf{q}^{k+1}) - f^*(\mathbf{q}^{k+1})$. □

Next, we show that we can always restart the DC algorithm from a computed stationary point.

Lemma 5 *Suppose that in Algorithm 2 the termination criterion in line 5 is met with supply* $\mathbf{s}$ *and price vector* $\mathbf{p}$. *If* $\mathbf{p}$ *is no equilibrium price, then there exists a descent direction* $\mathbf{e} \in \pm\{0,1\}^n$ *of the Lyapunov function at* $\mathbf{p}$. *If we restart the algorithm with* $\tilde{\mathbf{p}}^0 := \mathbf{p} + \mathbf{e}$, *we have for all elements* $(\tilde{\mathbf{p}}^k, \tilde{\mathbf{s}}^k)$ *of the new sequence that* $L(\tilde{\mathbf{p}}^k) \leq L(\mathbf{p}) - 1$.

Proof. If the returned price $\mathbf{p}$ is no equilibrium price, then it is no minimizer of the Lyapunov function (Ausubel, 2006). It follows by $L^\natural$-convexity of L that there exists $\mathbf{e} \in \pm\{0,1\}^n$ with $L(\mathbf{p}+\mathbf{e}) \leq L(\mathbf{p}) - 1$ (Murota, 2003). By Property 1 in Proposition 6 we have that $L(\tilde{\mathbf{p}}^k) \leq L(\mathbf{p}) - 1$ for the sequence of prices generated after the restart with initial price $\tilde{\mathbf{p}}^0 = \mathbf{p}+\mathbf{e}$. Since L possesses a minimizer (Ausubel, 2006) and after each restart the value of the Lyapunov function decreases by at least 1, the algorithm terminates with an equilibrium price. □

This completes the proof of Theorem 3: In each step of the main loop, the value of $L(\mathbf{p}^k)$ strictly decreases by an integer amount, and if we leave the main loop, we either restart with a price vector of a strictly smaller value, or we terminate, if we are at a global minimum already.

Lemma 6 *Suppose that the prices $\mathbf{p}^{k+1}$ in Step 4 of Algorithm 2 are chosen deterministically. Let $R \in \mathbb{Z}_{\geq 0}$ be the number of restarts of the algorithm and let $S_r = (\mathbf{s}_r^0, \mathbf{s}_r^1, \dots, \mathbf{s}_r^{|S_r|})$ the sequence of iterates generated in Step 3 after the r-th restart for $r = 0, \dots, R$ (S_0 is the sequence before the first restart). Then for $r_1 \neq r_2$, S_{r_1} and S_{r_2} do not contain any common bundle. Moreover, for each r the bundles $\mathbf{s}_r^0, \dots, \mathbf{s}_r^{|S_r|-1}$ are pairwise distinct.*

Proof. Suppose that $\mathbf{s}_{r_1}^k = \mathbf{s}_{r_2}^l$ for some $r_1 \leq r_2$ and k, l. Then we have for the computed prices in Step 4 that $\mathbf{p}_{r_1}^{k+1} = \mathbf{p}_{r_2}^{l+1}$, so $L(\mathbf{p}_{r_1}^{k+1}) = L(\mathbf{p}_{r_2}^{l+1})$. This can only happen if $r_1 = r_2$, since otherwise $L(\mathbf{p}_{r_2}^{l+1}) \leq L(\mathbf{p}_{r_1}^{k+1}) - 1$ by Lemma 5. Now suppose that $r_1 = r_2 = r$ and $k \leq l$. Then again $L(\mathbf{p}_r^{k+1}) = L(\mathbf{p}_r^{l+1})$. By Property 1 of Proposition 6, it follows that $k = l-1$ and the termination criterion is satisfied in iteration l, so $l = |S_r|$ and the bundles $\mathbf{s}_r^0, \dots, s_r^{|S_r|-1}$ are pairwise distinct. □

A.6 DC Algorithm

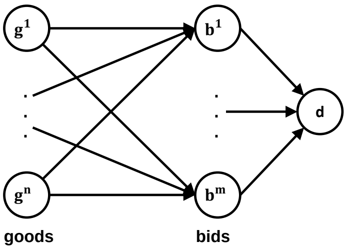

Fig. 6 A flow graph modeling (LP).

For the DC algorithm, reformulating the (LP) as a min-cost flow problem comes with a significant computational advantage as compared to solving it with a generic LP-solver. We briefly describe the general min-cost flow problem. For more details, we refer to Ahuja et al. (1993). Given a directed graph, an arc is a tuple $(\mathbf{v}, \mathbf{w})$ where $\mathbf{v}$ and $\mathbf{w}$ are nodes of the graph. We denote by $u(\mathbf{v}, \mathbf{w}) \geq 0$ the maximum capacity of this arc and by $c(\mathbf{v}, \mathbf{w}) \in \mathbb{R}$ the cost per unit flow along $(\mathbf{v}, \mathbf{w})$. For a node $\mathbf{v}$, we denote by $\beta(\mathbf{v}) \in \mathbb{R}$ the supply at node $\mathbf{v}$. Depending on the sign of $\beta(\mathbf{v})$, a total flow of $|\beta(\mathbf{v})|$ must leave (positive sign) or enter (negative sign) $\mathbf{v}$. If the supply is 0, the inflow must equal the outflow. A flow f assigns a value $f(\mathbf{v}, \mathbf{w}) \in \mathbb{R}$ to each arc, the amount of flow from $\mathbf{v}$ to $\mathbf{w}$. It is feasible, if $0 \leq f(\mathbf{v}, \mathbf{w}) \leq u(\mathbf{v}, \mathbf{w})$ for each arc $(\mathbf{v}, \mathbf{w})$ in the network, and $\sum_{\mathbf{w}} f(\mathbf{v}, \mathbf{w}) - \sum_{\mathbf{w}} f(\mathbf{w}, \mathbf{v}) = \beta(\mathbf{v})$ for all nodes $\mathbf{v}$, where the sums run over all $\mathbf{w}$ such that $(\mathbf{v}, \mathbf{w})$, respectively $(\mathbf{w}, \mathbf{v})$ is an arc in the network. The cost of the flow is equal to $\sum_{(\mathbf{v},\mathbf{w})} c(\mathbf{v},\mathbf{w})f(\mathbf{v},\mathbf{w})$. The objective of the min-cost flow problem is to find a feasible flow with minimal cost.

The linear program (LP) is used to solve Step 4 in Algorithm 2 where we need to compute a price vector $\mathbf{p}^{k+1}$ at which the bundle $\mathbf{t}+\mathbf{s}^k$ is demanded. A straightforward network flow model for (LP) is illustrated in Figure 6. For each good $i \in \{1,\dots,n\}$ there is a node $\mathbf{g}^i$, and for each of the $m = |\mathcal{B}_+|$ positive bids indexed by $j \in \{1,\dots,m\}$ there is a node $\mathbf{b}^j$. Finally, there is a destination node $\mathbf{d}$. In our flow network, there is an arc $(\mathbf{g}^i,\mathbf{b}^j)$ from each good i to each bid j with unlimited capacity $u(\mathbf{g}^i,\mathbf{b}^j) = \infty$ and cost $c(\mathbf{g}^j,\mathbf{b}^i) = -b^j_i$, i.e., the negative value of bid j for good i. The arcs $(\mathbf{b}^j,\mathbf{d})$ from the bids to the destination node have capacity $u(\mathbf{b}^j,\mathbf{d}) = w(\mathbf{b}^j)$ and cost $c(\mathbf{b}^j,\mathbf{d}) = 0$. In Step 4 of Algorithm 2, a supply of $\mathbf{t}+\mathbf{s}^k$ must be distributed among the bids. We set the supply of node $\mathbf{g}^i$ to $\beta(\mathbf{g}^i) = t_i + s^k_i$ and the supply of node $\mathbf{d}$ to $\beta(\mathbf{d}) = -\sum_{i=1}^n t_i + s^k_i$. Finally, the supply of node $\mathbf{b}^j$ is set to $\beta(\mathbf{b}^j) = 0$. Since $\mathbf{t}+\mathbf{s}^k \in \Delta_{W_+}$ (Appendix Lemma 4), $\sum_{i=1}^n \mathbf{t}_i + \mathbf{s}^k_i \le \sum_{j=1}^m w(\mathbf{b}^j)$, so a feasible flow f exists. Moreover, since the capacities and supplies are all integral, an integral optimal flow exists. Note that the proposed flow network contains arcs with negative cost. If required by a specific solver, it can however easily be transformed into a network with only non-negative costs (Ahuja et al., 1993, p. 40).

We assume that the applied min-cost flow solver provides us with an integral optimal flow f, as well as with an integral optimal dual solution, consisting of node potentials $\pi(\mathbf{v}) \in \mathbb{R}$ for each node $\mathbf{v}$ in the network. These satisfy the following complementary slackness conditions (Ahuja et al., 1993, Theorem 9.4).

1. If $c(\mathbf{v},\mathbf{w}) + \pi(\mathbf{v}) - \pi(\mathbf{w}) > 0$, then $f(\mathbf{v},\mathbf{w}) = 0$.
2. If $0 < f(\mathbf{v},\mathbf{w}) < u(\mathbf{v},\mathbf{w})$, then $c(\mathbf{v},\mathbf{w}) + \pi(\mathbf{v}) - \pi(\mathbf{w}) = 0$.
3. If $c(\mathbf{v},\mathbf{w}) + \pi(\mathbf{v}) - \pi(\mathbf{w}) < 0$, then $f(\mathbf{v},\mathbf{w}) = u(\mathbf{v},\mathbf{w})$.

From the complementary slackness conditions it is not hard to deduce that $\mathbf{p}$ defined by $p_i = \pi(\mathbf{g}^i) - \pi(\mathbf{d})$ is an equilibrium price vector for the supply $\mathbf{t}+\mathbf{s}^k$, so we can choose $\mathbf{p}^{k+1} = \mathbf{p}$ in Step 4 of the algorithm.

Let us finally consider Step 3 of Algorithm 2, where a bundle $\mathbf{s}^k$ must be chosen that is demanded at price $\mathbf{p}^k \in \mathbb{Z}^n$ in the negative auction. This is particularly easy to do in the Product-Mix Auction (see also Baldwin, Goldberg, Klemperer, and Lock (2019)): For each bid $\mathbf{b}$ in the negative auction, choose a bundle $\mathbf{s}(\mathbf{b}) \in D_{\mathbf{b}}(\mathbf{p})$. By Equations (1) and (2), this can be done in linear time in the number of different goods. Then set $\mathbf{s}^k = \sum_{\mathbf{b}\in|\mathcal{B}_-|} \mathbf{s}(\mathbf{b})$. In our implementation, we choose a bundle $\mathbf{s}^k$ which is a vertex of $D_{|\mathcal{B}_-|}(\mathbf{p})$. This can be achieved by suitably perturbing $\mathbf{p}$: Let $\mathbf{q} = \mathbf{p} + \Delta$ be a price such that $D_{|\mathcal{B}_-|}(\mathbf{q}) \cap D_{|\mathcal{B}_-|}(\mathbf{p}) \neq \emptyset$ and $|D_{|\mathcal{B}_-|}(\mathbf{q})| = 1$. For example, $\Delta = (\varepsilon, 2\varepsilon, \dots, n\varepsilon)$ works for $\varepsilon > 0$ small enough. Then simply choose the unique $\mathbf{s}(\mathbf{b}) \in D_{|\mathcal{B}_-|}(\mathbf{q})$ and construct $\mathbf{s}^k$ as above.

A.7 Generating Valid Bid Groups

Algorithm 4: Algorithm for generating valid groups of positive and negative bids, used for experiments.

```
Input: Random parameters $a_0, \ldots a_n \in \mathbb{Z}_{>0}$, a random permutation $\sigma \in S_n$,
       weight $w \in \mathbb{Z}_{>0}$, displacement parameter $\mathbf{c} \in \mathbb{Z}^n_{\geq 0}$.
Output: Group of 3 positive and 1 negative bids.
Generate 2 vectors $\mathbf{v}^1, \mathbf{v}^2 \in \mathbb{Z}^n_{\geq 0}$:
for $i \in \{1, 2\}$ do
    Set $v^i_j = 0$ for all $j$ with $\sigma(j) \leq 2$ and $\sigma(j) \neq i$.
    Set $v^i_j = a_j$ if $\sigma(j) = i$.
    Otherwise, choose $v^j_i \in \{0, a_j\}$ uniformly at random.
end for
Place positive bids at $\mathbf{v}^1, \mathbf{v}^2$.
Place a negative bid at the coefficient-wise maximum $\mathbf{v}^1 \wedge \mathbf{v}^2$.
Let $J = \{j : v^1_j \neq v^2_j\}$. Place a positive bid at $\mathbf{v}^1 \wedge \mathbf{v}^2 + a_0 \mathbf{e}_J$.
Assign weight $w$ to all these bids and shift them by $\mathbf{c}$.
```

A.8 Experimental Results

#pos. bids	#neg. bids	#goods	time DC (ms)	time SD (ms)
1020	20	10	31	394
1020	20	20	63	352
1020	20	30	105	410
1020	20	40	133	505
1020	20	50	206	665
1200	200	10	60	502
1200	200	20	157	478
1200	200	30	288	522
1200	200	40	453	620
1200	200	50	597	791
1500	500	10	128	649
1500	500	20	313	664
1500	500	30	562	682
1500	500	40	916	775
1500	500	50	1163	962
3020	20	10	115	1152
3020	20	20	252	1116
3020	20	30	445	1278
3020	20	40	592	1225
3020	20	50	970	1366
3200	200	10	175	1303
3200	200	20	413	1236
3200	200	30	628	1226
3200	200	40	1082	1421
3200	200	50	1649	1588
3500	500	10	244	1620
3500	500	20	606	1559
3500	500	30	983	1509
3500	500	40	1642	1575
3500	500	50	2803	1919

Table 2: Runtimes of the DC- and the steepest descent (SD)-algorithm. For each experimental setting, we generated 15 sample auctions. The indicated runtimes are the averages over the respective 15 samples.