Information Feedback and Efficiency in Multiattribute Double Auctions

Kevin M. Lochner[1] and Michael P. Wellman[2]

[1] RentMineOnline.com 1690 Bay Street, Suite 302, San Francisco, CA 94123, USA
kevin@rentmineonline.com
[2] University of Michigan, Computer Science & Engineering, Ann Arbor, MI 48109-2121, USA
wellman@umich.edu

Abstract. We investigate tradeoffs among expressiveness, operational cost, and economic efficiency for a class of multiattribute double-auction markets. To enable polynomial-time clearing and information feedback operations, we restrict the bidding language to a form of multiattribute OR-of-XOR expressions. We then consider implications of this restriction in environments where bidders' preferences lie within a strictly larger class, that of complement-free valuations. Using valuations derived from a supply chain scenario, we show that an iterative bidding protocol can overcome the limitations of this language restriction. We further introduce a metric characterizing the degree to which valuations violate the substitutes condition, theoretically known to guarantee efficiency, and present experimental evidence that the actual efficiency loss is proportional to this metric.

1 Introduction

Multiattribute auctions mediate the trade of goods defined by a set of underlying features, or *attributes*. Bids express offers to buy or sell *configurations* defined by specific attribute vectors, and the auction process dynamically determines both the transaction prices and the configurations of the resulting trades. Most research on multiattribute auctions addresses the single-good procurement setting, in which a single buyer negotiates with a group of candidate suppliers [3,5,8,13]. Extending to the two-sided case offers the potential for enhanced efficiency, price dissemination, and trade liquidity.

In mediating the trade of multiple goods, it is often beneficial to consider preferences for bundles of goods. The exponentially sized offer specifications induced by such *combinatorial* valuations present difficulties, both for expression of agent valuations [15] and computation of optimal allocations [14]. For certain subclasses of multi-unit valuations, however, these problems may be tractable. Notably, for valuations satisfying the *gross substitutes* condition, it is well known that a price equilibrium exists, and such equilibria support efficient allocations. Furthermore, *market-based algorithms*—distributed iterative procedures that search over a space of linear prices—reliably converge to equilibrium under gross substitutes. Market-based algorithms that rely exclusively on offers for individual goods in effect provide a polynomial scheme for approximate computation of efficient allocations. In Section 3, we review gross substitutes and its relation to syntactically defined bidder valuation classes.

S. Das et al. (Eds.): Amma 2009, LNICST 14, pp. 26–39, 2009.

Since a configuration in multiattribute negotiation corresponds to a unique type of good, the class of multi-unit valuations for multiattribute goods is equivalent to the class of combinatorial valuations. The problem of multi-unit multiattribute allocation therefore inherits the hardness results derived for combinatorial auctions, but moreover applied to a cardinality of goods that is itself exponential in the number of attributes.

In Section 4, we present a two-sided multiattribute auction admitting polynomial-time clearing given a restricted bidding language. We extend a previously developed clearing algorithm [9] with a polynomial-time information feedback algorithm, enabling the implementation of market-based algorithms. Our mechanism thus extends some of the efficiency results of combinatorial auctions to multiattribute domains. We provide evidence that the inclusion of information feedback to our auction design successfully compensates for the lack of expressive power of our bidding language.

Theoretical work is largely silent on the efficiency of market-based algorithms given valuations violating gross substitutes. In Section 5, we present natural ways in which complement-free valuations may violate the gross substitutes condition, invalidating the efficiency guarantee of market-based approaches. In an effort to quantify the expected performance limits of our mechanism against a larger class of valuations, we introduce a new metric on bidder valuations, based on the severity by which valuations violate gross substitutes. We apply this metric to a family of valuations, derived from a supply chain manufacturing scenario, and present simulation results demonstrating a correlation between our metric and expected market efficiency.

2 Auction Preliminaries

Auctions mediate the trade of goods among a set of self-interested participants, or *agents*, as a function of agent messages, or *bids*. In a *multiattribute* auction, goods are defined by vectors of *attributes*, $a = (a_1, \ldots, a_m)$, $a_j \in A_j$. A *configuration*, $x \in X$, is a particular attribute vector. Each configuration can be thought of as a unique type of good. An *allocation*, $g \in G$, is a multiset of such goods, that is, a set possibly containing more than one instance of a given configuration.

Bids define one or more *offers* to buy or sell goods. An offer pairs an allocation and a *reserve price*, (g, p), where $g \in G$ and $p \in \Re^+$. For a buy offer, the reserve price indicates the maximum payment a buyer is willing to make in exchange for the set of goods comprising allocation g. Similarly, the reserve price of a sell offer defines the minimum payment a seller is willing to receive to provide allocation g.

A *bid*, $b \in B$, defines a set of offers (often implicitly) which collectively define an agent's reserve price over the space of allocations. We use the term *valuation* to designate any mapping from the space of allocations to the positive real numbers: $v : G \mapsto \Re^+$, hence a bid defines a valuation. For ease of explication, we use the function $r : G \times B \mapsto \Re^+$ to indicate the reserve price of a bid for a given allocation. The *bidding language* of an auction defines the space B of expressible bids.

Each bidder maintains a single active bid: b_i for buyer i and b_j for seller j. To *clear* the market, the auction computes a *global allocation* comprising an assignment of individual allocations and associated payments. The computed allocation must be

1. *feasible*: the set of goods allocated to buyers is contained in the set of goods supplied by sellers, and the net payments are nonnegative, and
2. *acceptable*: individual payments meet the reserve price constraints expressed in the bids of buyers and sellers.

We assume agents have *quasilinear* preferences over alternative allocation and payment outcomes. Buyer i has quasilinear utility function $u_i(g, p) = v_i(g) + p$, where valuation v_i defines the net change in buyer utility when supplied with a given allocation, and p denotes the net payments made to the buyer. Similarly, seller j has utility function $u_j(g, p) = -v_j(g) + p$, where valuation v_j is interpreted as a cost function for supplying allocations.

The allocations and payments determine the realized utilities of all agents. To the extent that bids accurately reflect valuations, an auction can use bids as proxies for underlying valuations, and maximize the objective function for the valuations expressed through bids. The extent to which bids do not accurately reflect agent valuations may induce inefficient (suboptimal) global allocations, as the maximization employs an inaccurate objective function. A bidding language which is syntactically unable to fully convey agent valuations may therefore impede efficiency. Since the complexity of optimizing global allocation increases with the expressiveness of the bidding language, we face a general tradeoff between computational and allocational efficiency.

In a *direct revelation* mechanism, each agent submits at most a single bid, in the form of a valuation, without receiving any information about the bids of other agents. In *iterative auctions*, agents revise their bids over time based on summary information provided by the auction about the current auction state. Summary information is typically derived from the clearing algorithm given the current auction state, informing agents of their current hypothetical allocations as well as *price quotes* indicating the minimum or maximum prices to buy or sell allocations [16].

3 Allocation with Complement-Free Valuations

We start by revisiting complexity results for combinatorial allocation, focusing on *complement-free* bidder valuations. Non-complementarity assumptions are commonly invoked in economic models, including diminishing marginal utilities for consumers and decreasing returns to scale for producers, and the substitutes condition for Walrasian equilibria [12]. The class of complement-free buyer valuations contains all valuations which are never *superadditive* over configurations.

Definition 1. *A buyer valuation is complement-free (CF) if for any g_a and g_b,*

$$v(g_a) + v(g_b) \geq v(g_a \cup g_b).$$

A seller valuation (cost function) is complement-free it is not subadditive over configurations, that is, the direction of the above inequality is reversed for sellers.

It is known that no polynomial clearing algorithm can guarantee better than a 2-approximation for the general class *CF* [7]. In the sequel, we present subclasses of *CF* of increasing complexity, borrowing both terminology and complexity results from Lehmann et al. [10], with notation amended slightly for multiattribute domains (where unique goods correspond to the configurations).

3.1 Syntactic Valuation Classes

Syntactic valuations are built from *atomic valuations* and operators on those valuations.

Definition 2. *The atomic valuation* (x, p) *gives the value* p *to any allocation containing a unit of configuration* x*, and value zero to all other allocations.*

Definition 3. *Let* v_1 *and* v_2 *be two valuations defined on the space* G *of allocations. The valuations* $v_1 + v_2$ *(OR) and* $v_1 \oplus v_2$ *(XOR) are defined by:*

$$(v_1 + v_2)(g) = \max_{g' \subseteq g}(v_1(g') + v_2(g \setminus g')),$$
$$(v_1 \oplus v_2)(g) = \max(v_1(g), v_2(g)).$$

Informally, the valuation $(v_1 + v_2)(g)$ divides up allocation g among valuations v_1 and v_2 such that the sum of the resulting valuations is maximized. The valuation $(v_1 \oplus v_2)(g)$ gives the entire allocation to v_1 or v_2, depending on which values g higher.

Subclasses of complement-free valuations are derived by placing restrictions on how the *OR* and *XOR* operators may be combined. Class *OS* valuations use only the *OR* operator over atomic valuations, thereby expressing additive valuations. Class *XS* valuations apply *XOR* over atomic valuations, thereby expressing substitute valuations. Any valuation composed of *OR* and *XOR* (applied in arbitrary order) falls into class *XOS*. The best approximation factor that can be guaranteed for *XOS* valuations in polynomial time is bounded above by 2, and below by $\frac{4}{3}$ [7].

3.2 *OXS* Valuations

Definition 4. *Applying ORs over XS valuations yields a valuation in class OXS.*

For example, as a buy bid, the valuation $(x_1, p_1) + [(x_2, p_2) \oplus (x_3, p_3)]$ expresses a willingness to buy x_1 at a price of p_1, and independently expresses a willingness to buy either x_2 at a price of p_2, or x_3 at a price of p_3 (but not both), giving the following acceptable allocations:

$$\{(x_1, p_1), (x_2, p_2), (x_3, p_3), (\{x_1, x_3\}, p_1 + p_3), (\{x_1, x_2\}, p_1 + p_2)\}.$$

If all bids express *OXS* valuations, the clearing problem can be formulated as a polynomial-time bipartite matching problem [9].

3.3 Gross Substitutes

To define valuations exhibiting gross substitutability, we must first introduce the concept of a demand correspondence. The following definitions are with respect to buyers.

Definition 5. *Given valuation* v *and configuration prices* $\boldsymbol{p} = (p_{x_1}, \ldots, p_{x_n})$*, demand correspondence* $d(v \mid \boldsymbol{p})$ *denotes the set of allocations that maximize* $v(g) - \sum_{x \in g} p_x$.

Definition 6. *A valuation* v *is of class GS if for any price vectors* $\boldsymbol{p}$ *and* $\boldsymbol{q}$ *with* $p_i \leq q_i \ \forall i$ *and* $g_1 \in d(v \mid \boldsymbol{p})$*, there exists* $g_2 \in d(v \mid \boldsymbol{q})$ *such that* $\{x \in g_1 \mid p_x = q_x\} \subset g_2$.

Informally, *GS* requires that the demand for a given configuration be nondecreasing in the price of any other configuration. For sellers, the *supply* of a given configuration must be non*in*creasing in the price of others.

Valuations satisfying the gross substitutes condition admit efficiency through *market-based algorithms.* Such algorithms operate by iteratively providing agents with price quotes, requiring that agents express *demand sets* reflecting their optimal consumption or production choices at the given prices. Demand sets are expressible in any bidding language of complexity equal to or greater than class *OS*. Prices are adjusted at each iteration based on the relative supply and demand of each type of good, until the market reaches equilibrium. Computationally, market-based algorithms provide a fully polynomial approximation scheme, with complexity that is polynomial in the number of bidders, goods, and the inverse of the approximation factor [10].

4 Call Market Implementation

In this section, we present the bidding language and algorithms supporting our multiattribute call market implementation. Though we focus here on the discrete configuration-based bidding language employed in our experimental study, both the clearing and information feedback algorithms admit more general bid forms [9,11].

4.1 Bidding Language

As discussed, multiattribute goods are defined in terms of possible *configurations* assigning values to attributes. The simplest multiattribute bidding unit expresses a maximum/minimum price at which to trade a given quantity of a single configuration.

Definition 7 (Multiattribute Point). *A* multiattribute point *of the form* (x, p, q) *indicates a willingness to buy up to total quantity* q *of configuration* x *at a unit price no greater than* p *(for* $q > 0$*). A negative quantity (*$q < 0$*) indicates a willingness to sell up to* q *units at a price no less than* p*.*

Participants in multiattribute auctions often wish to express flexibility over alternative configurations. For example, a computer buyer may be willing to accept various possibilities for processor type/speed, memory type/size/speed, etc., but at configuration-dependent reserve prices.

Definition 8 (Multiattribute *XR* Unit). *A* multiattribute *XR* unit *is a triple of the form* $((x_1, \ldots, x_N), (p_1, \ldots, p_N), q)$*, indicating a willingness to trade any combination of configurations* $(x_1, \ldots, x_N)$ *at unit prices* $(p_1, \ldots, p_N)$ *up to total quantity* $|q|$*, where* $q > 0$ *indicates a buy offer,* $q < 0$ *a sell offer.*

For example, given *XR* unit $((x_1, x_2, x_3), (p_1, p_2, p_3), 4)$, the allocation $\{x_1, x_1, x_2\}$ would be acceptable at total payment not greater than $p_1 + p_1 + p_2$.

In a slight abuse of notation, let $r(XR, x) = p$ select the unit reserve price for configuration x in the specified *XR* unit. Note that a multiattribute point is equivalent to an *XR* unit with single configurations and prices. To simplify our examples, we use the multiattribute point notation when an *XR* unit includes exactly one configuration.

Our final language construct is an *OR* extension of the *XR* unit.

Definition 9 (Multiattribute *OXR* Bid). *A* multiattribute *OXR* bid, $\{XR_1, \ldots, XR_M\}$, *indicates a willingness to trade any combination of configurations such that the aggregate allocation and payments to the bidder can be divided among the XR units such that each* (g, p) *pair is consistent with its respective XR unit.*

The bidding language constructs presented here can be classified within the syntactic framework presented above. A multiattribute point (x, p, q) expresses the valuation

$$\underbrace{(x, p) + (x, p) + \cdots}_{\text{total of } |q| \text{ atomic elements}}.$$

The additional quantity designation in a multiattribute point provides compactness over the equivalent *OR* expression when valuations are linear in quantity. A multiattribute *XR* unit with quantity q defines the valuation

$$\underbrace{[(x_1, p_1) \oplus \cdots \oplus (x_N, p_N)] + [(x_1, p_1) \oplus \cdots \oplus (x_N, p_N)] + \cdots}_{\text{total of } |q| \text{ XOR elements}}.$$

The multiattribute *XR* unit is less expressive than the general class *OXS* because it defines an *OR* over a set of identical *XOR* expressions, thus imposing a constraint that valuations be linear in quantity, and *configuration parity*, that is, the quantity offered by a bid is configuration-independent [9]. The *OXR* class is equivalent in expressiveness to *OXS*, though multiattribute *OXR* bids can be more compact and computationally convenient to the extent that valuations are linear in quantity.

4.2 Clearing

Previous work [9] explored the connection between bidding languages and clearing algorithms for this domain. Here we provide the main results but present them for only the *OXR* bidding language employed in the current study. The result holds for more general conditions on the bidding language as described in the earlier paper.[1]

Clearing the market requires finding the global allocation that maximizes the total trade surplus, which is the *Global Multiattribute Allocation Problem* (*GMAP*). For a certain class of bids, which includes *OXR* bids, *GMAP* can be divided into two discrete steps: identifying optimal bilateral trades (the *Multiattribute Matching Problem*, *MMP*), then maximizing total surplus as a function of those trades.

In the case of *OXR* bids, the multiattribute matching problem determines the optimal configuration x to trade between each pair of buy and sell *XR* units. For buy *XR* unit $XR_b = (\mathit{configs}^b, \mathit{prices}^b, q^b)$ and sell *XR* unit $XR_s = (\mathit{configs}^s, \mathit{prices}^s, q^s)$,

$$MMP_x(XR_b, XR_s) = \operatorname*{argmax}_{x \in X}[r(XR_b, x) - r(XR_s, x)]. \quad (1)$$

The value achieved by the multiattribute matching solution (1) is called the *MMP surplus*, $MMP_s(XR_b, XR_s)$.

[1] The earlier paper [9] characterized bidding languages in terms of allocation constraints, rather than the complement-free hierarchy employed in the present work.

Define *BX* as the set of all *XR* units contained in the buyers' *OXR* bids, and *SX* the set of all *XR* units in the sellers' *OXR* bids. We start by solving *MMP* (1) for each pair in $BX \times SX$. *GMAP* is then formulated as a network flow problem, specifically the *transportation problem*, with source nodes *SX*, sink nodes *BX*, and link surplus (equivalently, negative link costs) equal to the values of MMP_s on $BX \times SX$. The optimal solution flow along a given link designates a quantity traded between the traders whose bids contain the respective *XR* units, and the configuration to be traded is the solution to MMP_x between the *XR* units.

4.3 Information Feedback

The decomposition of *GMAP* into *MMP* and subsequent network optimization can also be exploited for computing price quotes. To calculate quotes, we first find the required surplus (i.e., solution to MMP_s) for a new trade with a particular trader to be included in the efficient set. We can then determine the required price offer to that trader for any available configuration as a function of that required surplus. The computed price will be the quote for a $(configuration, trader)$ pair; taking the min/max over all sellers/buyers yields the ask/bid quote for a configuration.

This process is best described through example. Figure 1 depicts the *GMAP* formulation for a set of three sell offers (shown at left) and three buy offers (shown at right), all expressed as *XR* units. For example, XR_1 is an offer to sell a unit of either x_1 or x_2 at a price of 11. The solutions to MMP_s are indicated on the links connecting pairs of offers. The solution to *GMAP* is indicated by the bold links, in this case XR_5 and XR_2 trading one unit of x_2, and XR_3 and XR_6 trading one unit of x_2.

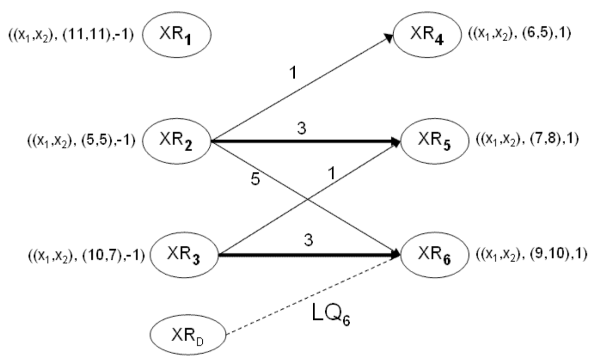

Fig. 1. *GMAP* formulation with three sell offers (left) and three buy offers (right). The optimal solution is indicated in bold. XR_D is a dummy node added for computing a link quote.

We now calculate the bid quote for x_2. As depicted in Figure 1, we first connect a dummy node (XR_D) to one of the existing buy nodes (node XR_6). We now calculate the minimum link surplus on the new edge that would increase the value of the optimal network flow. The computed link quote, LQ_6, is the trade surplus ($MMP_s(XR_6, XR_D)$) required for a new bid to trade with node XR_6. The link quote for each buy node must be calculated, producing a link quote for each $XR_k \in BX$.

The bid quote for a given configuration x is then:

$$\max_k(r(XR_k, x) - LQ_k).$$

Continuing with the example, Figure 2 depicts the computed link quotes. In this instance, the bid quote for configuration x_2 with XR_6 would be the offered price of 10, less the required link surplus of 4, producing a quote of 6 to transact *with that unit*. The bid quote for the configuration is the maximum over all the units, which is also 6. An offer price of 6 for x_2 would be sufficient to trade with either XR_6 or XR_5, as the quoted price for XR_5 would also be 6 (with a reserve price of 8 and required link surplus of 2).

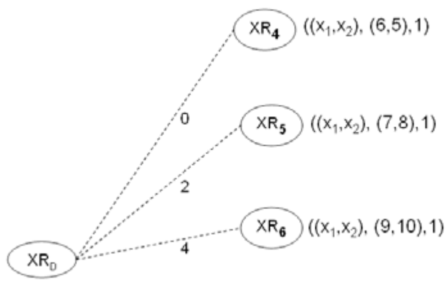

Fig. 2. Link quotes computed for a bid quote given the *GMAP* formulation of Figure 1

Finally, as confirmation that this process has produced a valid quote, we can consider the outcome of a new sell offer for a unit of x_2 at the quoted price. Figure 3 depicts this situation for the case that the new bid transacts with XR_5 (the algorithm will break the tie randomly) and shows that inclusion of the new bid has increased the trade surplus by 1 to a total of 7.

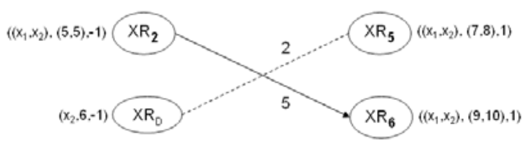

Fig. 3. *GMAP* solution for Figure 1 with a new sell offer at the quoted price

It is apparent from this formulation that once all link quotes have been determined, computing configuration quotes is proportional to the number of *XR* units. This implies that the complexity of a single configuration quote is invariant to the size of attribute space when the *GMAP-MMP* decomposition is applicable. Although computing quotes for all configurations entails complexity linear in the number of configurations, a bidder-driven query process for configuration quotes may still support market-based algorithm efficiency in large or continuous attribute domains.

Computing link quotes on the network flow graph is also achievable in polynomial time, using a specialization of the cycle-canceling algorithm [1]. Given that computation of a link quote requires perturbing the optimal network flow by quantity of only a single unit, the cycle-canceling algorithm can be adapted to a shortest-path algorithm, where an all-pairs shortest-path algorithm computes all required link quotes with complexity polynomial in the number of *XR* units. In practice, we require two iterations of the shortest-path algorithm, one iteration each for bid quotes and ask quotes.

5 Multiattribute Valuations

Our call market supports the direct expression of *OXS* valuations. However, many valuations natural for multiattribute domains fall outside of class *OXS*. For example, it is commonly desirable to ensure *homogeneity*, where all configurations in an allocation share values on one or more attributes [4]. Valuations placing higher values on homogeneous allocations are expressible with an *XOS* bidding language but not *OXS*.

The following example, inspired by a supply chain trading scenario [2], illustrates another situation where seller valuations fall outside of class *OXS*, and may violate *GS*.

Example 1. PCs are built from two components: cpu and memory. Assume that a manufacturer has one unit of $cpu = fast$, one unit of $cpu = slow$, one unit each for $memory \in \{large, medium, small\}$, with the following allowable configurations:

1. configuration x_1: $\{fast, large\}$
2. configuration x_2: $\{fast, medium\}$
3. configuration x_3: $\{slow, small\}$
4. configuration x_4: $\{slow, medium\}$

The production possibilities are then $\{x_1, x_4\}$, $\{x_1, x_3\}$, and $\{x_2, x_3\}$. The induced seller valuation is not expressible using an *OXS* language. The nearest *OXR* bid approximations require the seller to either overstate (bid B1) or understate (bids B2 and B3) his production capabilities:

$$(((x_1, x_2), (p_1, p_2), -1), ((x_3, x_4), (p_3, p_4), -1)) \tag{B1}$$

$$((x_1, p_1, -1), ((x_3, x_4), (p_3, p_4), -1)) \tag{B2}$$

$$(((x_1, x_2), (p_1, p_2), -1), (x_3, p_3, -1)) \tag{B3}$$

Assume that within the above production possibilities, the seller has a unit cost of 3 for all configurations, with total cost additive in unit cost. The exact *XOS* valuation would be

$$((x_1, 3) + (x_4, 3)) \oplus ((x_1, 3) + (x_3, 3)) \oplus ((x_2, 3) + (x_3, 3)).$$

This valuation is also not in class *GS*. Assume the prices of x_1 and x_4 are 5, and x_2 and x_3 are priced at 4. At these prices, the optimal production bundle is (x_1, x_4) which yields a surplus of 4. If the price of x_1 drops to zero, the optimal production bundle becomes (x_2, x_3), yielding a surplus of 2. Hence, the supply of x_4 decreases with a decrease in the price of x_1, which violates the gross substitutes condition for sellers.

6 A New Valuation Metric

Despite the limited expressive power of *OXS* bidding, we expect the iterative (market-based) version of our multiattribute auction to allocate effectively as long as valuations satisfy *GS*, or nearly do. To better characterize these situations, we introduce a measure of the *degree* to which a valuation violates the *GS* conditions.

6.1 Gross Substitutes Revisited

As defined above, *GS* requires that the demand for goods be nondecreasing in the prices of other goods. Intuitively, a price adjustment process will ultimately reach equilibrium if a price perturbation intended to reduce (increase) the demand of over(under)-demanded goods does not reduce (increase) the demand for other goods.

For valuation v satisfying *GS*, the demand correspondence condition holds for all price vectors and perturbations. Formally, given $d(v \mid \boldsymbol{p})$, the set of allocations maximizing $v(g) - \sum_{x \in g} p_x$, for all vectors of configuration prices $\boldsymbol{p} = (p_{x_1}, \ldots, p_{x_n}) \in \Re_+^n$, and all single price perturbations $\boldsymbol{dp} \in \Re_+^n$, for any $g_1 \in d(v \mid \boldsymbol{p})$ there exists $g_2 \in d(v \mid \boldsymbol{p} + \boldsymbol{dp})$ such that $\{x \in g_1 \mid dp_x = 0\} \subset g_2$.

6.2 Gross Substitutes Violation

Let $\boldsymbol{p}, \boldsymbol{dp} \in \Re_+^n$, with $g_i \in d(v \mid \boldsymbol{p})$. The *gross substitutes violation* is given by:

$$GSV(v, \boldsymbol{p}, \boldsymbol{dp}, g_i) = \min_{g \in d(v \mid \boldsymbol{p} + \boldsymbol{dp})} |\{x \in g_i \mid dp_x = 0\} \setminus \{x \in g \mid dp_x = 0\}|.$$

Intuitively, this measure counts the *number* of violations of the *GS* condition for a specific initial price vector and price change. Valuations satisfying *GS* have a violation count of zero for all initial prices, demand sets, and perturbations. Valuations that do not satisfy *GS* will have positive values of *GSV* for one or more combinations of $(\boldsymbol{p}, \boldsymbol{dp}, g)$.

To simplify the exposition hereon, we assume $d(v \mid \boldsymbol{p})$ maps to a single g for any $\boldsymbol{p}$, and use $x \in d(v \mid \boldsymbol{p})$ to indicate a good from that demand set. We next define the gross substitutes violation for a valuation and a price vector as the average *GSV* over all minimal single-price perturbations that ensure a new demand set.

$$GSV(v, \boldsymbol{p}) = \frac{1}{n} \sum_{i=1}^{n} GSV(v, \boldsymbol{p}, \boldsymbol{dp}^i, d(v \mid \boldsymbol{p})),$$

where $\boldsymbol{dp}^i = (0, \ldots, 0, dp_i, 0, \ldots, 0)$, and

$$dp_i = \min_{dp} dp \text{ s.t. } d(v \mid \boldsymbol{p}) \neq d(v \mid (p_1, \ldots, p_i + dp, \ldots, p_n)).$$

Next, define the *expected* gross substitutes violation for a valuation as the expected value of *GSV* for random $\boldsymbol{p}$ ($p_i \sim U[0, \bar{p}]$),

$$EGSV(v) = E[GSV(v, \boldsymbol{p})].$$

The intuition behind using the expected *GSV* of a valuation (the average, rather than the maximum or minimum) is that any given run of a market-based algorithm traces a particular trajectory in price space, and the average violation is a proxy for the probability of seeing any specific violation.

7 Testing the *EGSV*-Efficiency Relationship

When *GS* holds, *EGSV* is zero, and market-based algorithms achieve full efficiency. Our hypothesis is that when the condition fails, realized efficiency will be decreasing in *EGSV*, all else equal. To evaluate this hypothesis, we employed a component-based model of configurations, as in Example 1. In this model, valuation complexity is determined by the *configuration structure*, as well as by the respective inventory levels and component costs of sellers.

For example, a valuation defined over configurations $\{x_1, x_2, x_3\}$ will violate *GS* to the extent that swapping production from one configuration to another requires additional components that are allocated to the third. Treating configurations $\{x_1, x_2, x_3\}$ as sets of components, assume that switching production from x_1 to x_2 requires additional components $x_2 \setminus x_1$. If an agent has no additional inventory of the components $(x_2 \setminus x_1) \cap x_3$ then the induced valuation will have a *GSV* of 1 for some price levels. In this way, variation both in the composition of configurations and the inventory levels of agents induces different levels of substitutability in agent valuations.

In the example above, if $x_2 \setminus x_1$ included two distinct components used by two different configurations, then the bidder valuation would have $GSV = 2$ for some price vectors, and thus a nonzero *EGSV* value. Conversely, if an agent had excess inventory of $x_2 \setminus x_1$, then the induced valuation would have $GSV = 0$ for all prices, and therefore the valuation would have an *EGSV* value of zero.

7.1 Valuation Generation

We generate a configuration structure by constructing random configurations until we have 20 distinct instances. For each configuration, we probabilistically include any one of eight unique components in the configuration (i.e., configurations may have variable numbers of components), while additionally requiring that any single configuration have at least three components. Given this structure, we randomly sample costs and inventory to generate a seller valuation. Seller inventories for each component are drawn independently from the discrete uniform distribution $[0, 3]$, while seller costs per component are drawn from the discrete uniform distribution $[30, 80]$.

We then evaluate *EGSV* for the induced valuation with respect to the price distribution from which agent valuations are drawn. For each price sample $\boldsymbol{p}$,

1. determine the optimal production set $g^* = d(v \mid \boldsymbol{p})$,
2. identify all minimal single-price changes sufficient to change g^*, and
3. sum the *GS* valuations over these perturbations.

We iterate this process with random price samples until the standard error of *EGSV* is below .05. We generated a set of 100 valuations for each configuration structure, recording the costs and inventory, along with the *EGSV* value for each such valuation. We generated and evaluated seller valuations for 277 configuration structures, yielding a total of 27700 seller valuations.

7.2 Market Simulation

Each problem instance comprises 10 buyers and 10 sellers. For each configuration structure, we first sort the set of 100 generated seller valuations by *EGSV*. We define a unique problem instance for each contiguous set of 10 seller valuations, using the previously generated inventories and costs for each valuation, and taking the average *EGSV* value (denoted *aGSV*) of the 10 sellers to classify the problem instance. We thus generate 90 problem instances for each configuration structure.

We randomly generate buyer valuations for each problem instance. Each buyer has demand for two units, with full substitutability (i.e., will accept any combination of two goods at their reserve prices). Buyer reserve prices are drawn from the discrete uniform distribution $[400, 500]$ for each configuration.

For each problem instance, we first solve the allocation problem to determine the maximum achievable surplus. We then simulate bidding until quiescence, computing the fraction of efficiency achieved. To quantify the benefit of information feedback, we take the first iteration of bidding as the direct-revelation outcome. To evaluate the benefit of direct expression of substitutes—as in the *OXS* class supported by the *OXR* bidding language—we repeated the simulation with a class *OS* bidding language. Each problem instance thus produces four data points: one for each of $(direct, iterative) \times (OS, OXR)$.

Agents employ myopic best-response bidding, offering their true values at each iteration for a profit-maximizing set of goods. Given that the bidding language cannot fully express seller valuations, sellers are forced to approximate. To generate an *OS* bid, sellers find the feasible production bundle that maximizes profit at current prices (assuming a default price when quotes are not available). To generate an optimal *OXR* bid, sellers start with the optimal *OS* bid, and expand this to a feasible *OXR* bid.

7.3 Simulation Results

We aggregated the simulation results over all configuration structures and sorted the data by aGSV value into 10 bins. Figure 4 plots the average achieved fraction of

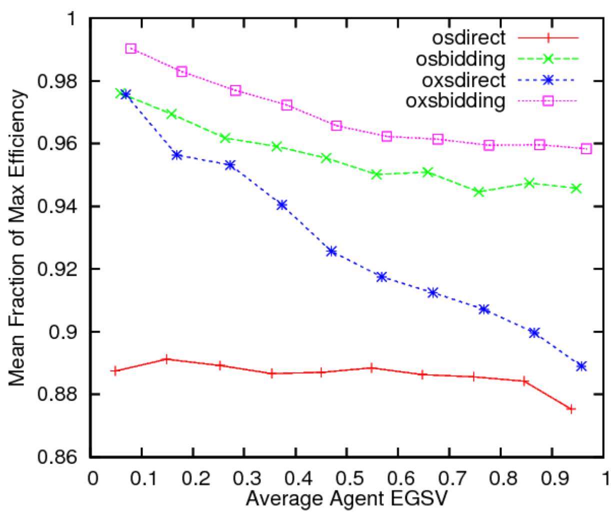

Fig. 4. Mean efficiency for average realized *EGSV*

maximal surplus as a function of aGSV value, for both direct-revelation and iterative mechanisms, for both the *OS* and *OXR* bidding languages.

For aGSV values close to zero, the substitutes condition is nearly satisfied for all valuations. Figure 4 confirms that iterative mechanisms perform well in this situation, averaging more than 97% efficiency for both *OXR* and *OS* bidding languages. The direct *OXR* mechanism (but not direct *OS*) also achieves this level of efficiency for low aGSV values. We conjecture that the majority of low *EGSV* valuations were also in class *OXS*, and therefore the ability to express substitutability through *OXR* bids is sufficient to achieve effective allocations without iteration.

Notable in Figure 4 is that the iterative mechanisms outperform the direct *OXR* mechanism by a margin that increases in aGSV value. We suspect this reflects valuations deviating further from class *OXS* with higher *EGSV* values. Despite increasing valuation complexity, the iterative mechanisms maintain a high level of efficiency, falling only to 95% as aGSV values reach 1. In this setting, information feedback is able to compensate for the lack of expressive power of a class *OXS* bidding language.

Finally, we observe that the iterative *OXR* mechanism outperforms the *OS* mechanism over all aGSV values. We hypothesize that the direct expression of substitutes allows the market-based algorithm to escape local maxima, as our mechanism does not implement a provably convergent market-based algorithm for *OS* bids.

8 Conclusions

We have introduced an implemented multiattribute call market with polynomial-time clearing and information feedback operations for a bidding language supporting a restricted class of combinatorial valuations. To our knowledge, this is the first call market of its kind presented in literature.

We analyzed the expected efficiency of our mechanism from the perspective of known hardness results derived for combinatorial auction settings, given complement-free bidder valuations. Using information feedback, iterative market-based algorithms can achieve efficient allocations given valuations satisfying the gross substitutes condition. Moreover, in some cases, iterative bidding can successfully compensate for expressive deficiencies imposed by a restricted bidding language.

Finally, we presented a new metric on bidder valuations, derived from the ways in which valuations violate *GS*. Experimental trials produce evidence that this metric correlates with the expected efficiency of market-based algorithms. The results suggest that measuring the degree of *GS* violation may provide a useful guide for predicting the performance of iterative bidding mechanisms, beyond the scope of environments for which theoretical guarantees apply.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Englewood Cliffs (1993)
2. Arunachalam, R., Sadeh, N.M.: The supply chain trading agent competition. Electronic Commerce Research and Applications 4, 63–81 (2005)

3. Bichler, M.: The Future of e-Markets: Multidimensional Market Mechanisms. Cambridge University Press, Cambridge (2001)
4. Bichler, M., Kalagnanam, J.: Configurable offers and winner determination in multi-attribute auctions. European Journal of Operational Research 160, 380–394 (2005)
5. Che, Y.-K.: Design competition through multidimensional auctions. RAND Journal of Economics 24, 668–680 (1993)
6. Cramton, P., Shoham, Y., Steinberg, R. (eds.): Combinatorial Auctions. MIT Press, Cambridge (2005)
7. Dobzinski, S., Nisan, N., Schapira, M.: Approximation algorithms for combinatorial auctions with complement-free bidders. In: Thirty-Seventh Annual ACM Symposium on Theory of Computing, pp. 610–618 (2005)
8. Engel, Y., Wellman, M.P.: Generalized value decomposition and structured multiattribute auctions. In: Eighth ACM Conference on Electronic Commerce, San Diego, pp. 227–236 (2007)
9. Engel, Y., Wellman, M.P., Lochner, K.M.: Bid expressiveness and clearing algorithms in multiattribute double auctions. In: Seventh ACM Conference on Electronic Commerce, pp. 110–119. Ann Arbor, MI (2006)
10. Lehmann, B., Lehmann, D., Nisan, N.: Combinatorial auctions with decreasing marginal utilities. Games and Economic Behavior 55, 270–296 (2006)
11. Lochner, K.M.: Multiattribute Call Markets. PhD thesis, University of Michigan (2008)
12. Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, New York (1995)
13. Parkes, D.C., Kalagnanam, J.: Models for iterative multiattribute procurement auctions. Management Science 51, 435–451 (2005)
14. Sandholm, T.: The winner determination problem. In: Cramton, et al [6]
15. Segal, I.: The communication requirements of combinatorial allocation problems. In: Cramton, et al [6]
16. Wurman, P.R., Wellman, M.P., Walsh, W.E.: A parametrization of the auction design space. Games and Economic Behavior 35, 304–338 (2001)