# Information Feedback and Efficiency in Multiattribute Double Auctions

**Kevin M. Lochner** and **Michael P. Wellman**
University of Michigan
Computer Science & Engineering
Ann Arbor, MI 48109-2121 USA

**Abstract**

Studies of bidding languages for combinatorial auctions have highlighted tradeoffs between expressiveness and complexity of representation and computation of standard auction functions. When goods are *multiattribute*, the cost of supporting multi-unit offers is especially acute, since the underlying good space is itself exponential in the number of attribute dimensions. 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 the implications for this language restriction in environments where bidders' preferences lie within a strictly larger class, that of complement-free valuations. Using a family of multi-unit multiattribute valuations derived from a supply chain manufacturing scenario, we show that an iterative bidding protocol can often 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 degree of substitutes violation.

## 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. The majority of research into multiattribute auctions addresses the single-good procurement setting, in which a single buyer negotiates the purchase of a good from among a group of potential suppliers (Branco, 1997; Che, 1997; Parkes and Kalagnanam, 2005; Bichler, Kaukal, and Segev, 1999; Sunderam and Parkes, 2003). In the spirit of financial exchanges, two-sided markets for multiattribute goods offer the opportunity for enhanced efficiency, price dissemination, and trade liquidity.

In order to mediate the trade of multiple goods simultaneously, it is often beneficial to consider preferences covering combinations of such goods. For example, if an agent may value good A and B together for $10, but have no value for either good in isolation, an auction will typically need to support the expression of preferences over the space of combined allocations in order to produce efficient outcomes. Auctions that admit offers specifying such combinations are called *combinatorial*.



The exponentially sized offer specifications induced by combinatorial valuations present a particularly hard allocation problem, both in the expression of agent valuations (Segal, 2005) and in the algorithmic problem of computing optimal allocations (Sandholm, 2005; Sandholm et al., 2002). For certain subclasses of multi-unit valuations, however, the computation of efficient outcomes is made tractable. Notably, for valuations satisfying the *gross substitutes* condition, it is well known that a Walrasian equilibrium exists, and a market-based algorithm admitting offers only on individual goods can provide a fully polynomial approximation scheme for the computation of efficient allocations. In the Section "Allocation with Complement-Free Valuations", we review gross substitutes and its relation to syntactically defined bidder valuation classes.

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. The translation of combinatorial auction algorithms to multiattribute domains thus presents a new and challenging problem, as such algorithms typically assume (at least for practical purposes) a predefined and modest-sized set of goods.

In the Section "Call Market Implementation", we present a two-sided multiattribute auction admitting polynomial-time clearing given a restricted bidding language. We extend a previously developed clearing algorithm 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 the Section “Multiattribute Valuations”, 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.

## 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, where 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 of each type. A multiset of goods can be formally defined as a pairing of an underlying *set* of goods, and a *function* mapping that set to the positive integers:

$$g = (N, m) | N \subseteq X \wedge Q : N \mapsto \mathbb{Z}^+,$$

where for any $x$, $Q(x)$ defines the quantity of $x$. We use $Q_g(x)$ to denote the quantity of good $x$ for allocation $g$.

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 syntax of allowable bids, thereby defining the space $B$ of expressible bids.

We divide agents into buyers $C = \{1, \ldots, i, \ldots, c\}$ and sellers $S = \{c+1, \ldots, j, \ldots, c+s\}$.[1] Each bidder has a single bid, $b_i$ for buyer $i$ and $b_j$ for seller $j$. Upon receiving a new or revised bid, the auction determines whether the bid is *admissible* given its current state. If admissible, the bid is added to the *order book*, $\Omega$, of the auction, comprising the collection of all active buy and sell bids:

$$\Omega = \{\Omega^b, \Omega^s\} = \{\{b_1, b_2, \ldots, b_c\}, \{b_{c+1}, \ldots, b_{c+s}\}\}.$$

When an auction determines the allocations and payments of participants, the process is referred to as *clearing*. In a clear operation, the auction computes a *global allocation* $\{\Theta^b, \Theta^s\}$ comprising an assignment of individual allocations and associated payments:

$$\{\{\theta_1^b, \theta_2^b, \ldots, \theta_c^b\}, \{\theta_{c+1}^s, \ldots, \theta_{c+s}^s\}\},$$

where $\theta_i^b = (g_i, p_i)$ defines an allocation $g_i$ supplied to buyer $i$ in exchange for payment $p_i$, and $\theta_j^s = (g_j, p_j)$ defines an allocation of $g_j$ supplied by seller $j$, who receives payment $p_j$.

A global allocation is *feasible* if the set of goods allocated to buyers is contained in the set of goods supplied by sellers, and the net payments are non-negative.

$$\text{feasible}(\Theta^b, \Theta^s) \iff \begin{cases} \forall x \in X, \quad \sum_{i \in C} Q_{g_i}(x) \leq \sum_{j \in S} Q_{g_j}(x) \\ \sum_{(g_i,p_i) \in \Theta^b} p_i - \sum_{(g_j,p_j) \in \Theta^s} p_j \ \geq 0 \end{cases}$$

A global allocation is *acceptable* if individual payments meet the reserve price constraints expressed in the bids of buyers and sellers.

$$\text{acceptable}(\Theta^b, \Theta^s | \Omega) \iff \begin{cases} \forall (g_i, p_i) \in \Theta^b, r(g_i, b_i) \geq p_i \\ \forall (g_j, p_j) \in \Theta^s, r(g_j, b_j) \leq p_j \end{cases}$$

We can now formalize the clear operation as computing a feasible and acceptable global allocation based on the order book. There typically will be multiple global allocations which are both feasible and acceptable. The auction selects one such allocation based on its *clearing policy*, which defines the timing and implementation of the clear operation as a function of the auction state.

We assume agents have preferences over alternative allocation and payment outcomes which can be represented with *quasilinear* utility functions, meaning that utility is linear in payments. Buyer $i$ then 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 $p$ defines 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 bidder allocations and payments determine the realized utilities of all agents. If a clear operation maximizes the sum of all realized agent utilities, that is, the *global surplus*, we call the auction *efficient*. An obstacle in determining an efficient global allocation is that the auction must compute agent allocations without direct observation of the agent valuations. An intermediate goal of the auction process is therefore to elicit agent preference information. This elicitation happens by way of the agent bids. In signifying willing deals, bids place bounds on the agent valuations.

[1] This assumption precludes settings in which agents wish to simultaneously buy and sell goods. We could accommodate such agents to some extent by allowing them to bid under separate buyer and seller identities.

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. To the extent that bids accurately reflect valuations, a direct-revelation 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 suboptimal global allocations, as the optimization procedure is performed over an inaccurate objective function. A bidding language which is syntactically unable to fully convey agent valuations may therefore induce natural efficiency limitations in a direct-revelation mechanism. Importantly, the computational complexity of identifying an efficient global allocation increases with the expressiveness of the bidding language. This creates a natural tension between computational complexity and auction efficiency in many settings.

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 (Wurman, Wellman, and Walsh, 2001). Iterative mechanisms support the implementation of market-based algorithms, which can augment the range of bidder valuations admitting efficiency for a given bidding language.

## Allocation with Complement-Free Valuations

Before devoting attention to multi-unit multiattribute allocation, it is instructive to revisit complexity results for the more simple setting of combinatorial allocation. We restrict attention to *complement-free* bidder valuations. Non-complementarity assumptions are common in Economics, including diminishing marginal utilities for consumers and decreasing returns to scale for producers (Mas-Colell, Whinston, and Green, 1995).

The class of complement-free buyer valuations contains all valuations which are not *superadditive* over configurations.

**Definition 1** *A buyer valuation is complement-free (CF) if for any two allocations $g_a$ 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* (Dobzinski, Nisan, and Schapira, 2005). In the following subsections, we present subclasses of *CF* of increasing complexity (Lehmann, Lehmann, and Nisan, 2006), providing known efficiency bounds for polynomial-time allocation.[2]

### 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.*

Next, define the operators *OR* and *XOR* over valuations as follows:

**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:*

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

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 are created using only the *OR* operator over atomic valuations, and allow for the expression of additive valuations. Class *XS* valuations are created by applying the *XOR* operator over atomic valuations, and allow for the expression of substitute valuations. Any valuation composed of *OR* and *XOR* (applied in an arbitrary order) falls into class *XOS*, and is expressible by applying the *XOR* operator over *OS* valuations. The best approximation factor that can be guaranteed for *XOS* valuations in polynomial time is known to be bounded above by 2, and bounded below by $\frac{4}{3}$ (Dobzinski, Nisan, and Schapira, 2005).

### *OXS* Valuations

Applying the *OR* operator over *XS* valuations yields valuations of class *OXS*.

**Definition 4** *A valuation is OXS if expressible through the application of OR operators over XS valuations.*

For example, as a buy bid, the valuation

$$(x_1, p_1) + ((x_2, p_2) \oplus (x_3, p_3))$$

expresses the willingness to buy either $x_1$ 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:

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

If all valuations are of class *OXS*, the clearing problem can be formulated as a polynomial-time bipartite matching problem given class *OXS* bids.

[2] In the multiattribute setting, unique goods correspond to the configurations. We borrow both notation and analytical complexity results from Lehmann et al. in this section, with notation amended slightly for multiattribute domains.

## Gross Substitutes

To define valuations exhibiting gross substitutability, we first define an agent demand correspondence with respect to valuations and prices. The following definitions are with respect to buyers.

**Definition 5** *Given valuation $v$ and vector of configuration prices $\vec{p} = (p_{x_1}, p_{x_2}, \ldots, p_{x_n})$, the demand correspondence $d(v|\vec{p})$ maps to the set of allocations which maximize $v(g) - \sum_{x \in g} p_x$.*

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

Informally, the gross substitutes condition for buyers states that the demand for a given configuration is nondecreasing in the price of any other configuration. For sellers, the condition changes so that the supply of a given configuration is nonincreasing in the price of other configurations.

Valuations satisfying the gross substitutes condition admit efficiency through *market-based algorithms.* Market-based algorithms derive from the ideas of *general equilibrium theory* (Arrow, Block, and Hurwicz, 1959), under which markets simultaneously maximize efficiency and achieve a perfect balance of supply and demand, given profit-maximizing behavior of market participants. The condition of gross substitutes has been identified in a number of settings as sufficient to guarantee existence of a Walrasian equilibrium (Arrow, Block, and Hurwicz, 1959; Kelso and Crawford, 1982; Gul and Stacchetti, 1999; Milgrom and Strulovici, 2006).

Market-based 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 (Lehmann, Lehmann, and Nisan, 2006).

# Call Market Implementation

In this section, we present the bidding language and algorithms supporting our multiattribute call market implementation, and present complexity results for this mechanism more precisely. Although we present only the discrete configuration-based bidding language employed in our simulations, both the clearing and information feedback algorithms presented here admit the more general bid forms described by Engel, Wellman, and Lochner (2006).

## Bidding Language

As discussed, goods are assumed to be defined by a set of *attributes*, where a given instantiation of attributes designates a *configuration*. The most simple 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 $q > 0$). Conversely, negative quantity ($q < 0$) would indicate a willingness to sell up to $q$ units at a price no less than $p$.*

Participants in multiattribute auctions often wish to buy or sell one of several alternative configurations. This would happen, for example, if a buyer wishes to procure computers, and is willing to accept multiple alternatives with respect to attributes such as processor type/speed, memory type/size/speed, etc., but has a configuration-dependent reserve price.

**Definition 8 (Multiattribute *XR* Unit)** *A* multiattribute *XR* unit *is a triple $(configs, prices, quantity)$ of the form $((x_1, x_2, \ldots, x_N), (p_1, p_2, \ldots, p_N), q)$, indicating a willingness to trade any combination of configurations $(x_1, x_2, \ldots, x_N)$ at unit prices $(p_1, p_2, \ldots, p_N)$ up to total quantity $|q|$, where $q > 0$ indicates a buy offer, $q < 0$ indicates a sell offer.*

An *XR* unit with positive (negative) quantity expresses a willingness to accept (provide) any allocation of total quantity not greater than $|q|$, given that the total payment is not greater (less) than the sum of the unit prices expressed. 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, we define the $r$ operator over an *XR* unit and a configuration to denote the reserve price for the given configuration in the *XR* unit, that is, $r(XR, x) = p$ selects the unit reserve price for configuration $x$ in the *XR* unit. Note that a Multiattribute Point is equivalent to an *XR* unit with 1-tuple configurations and prices. To simplify the syntax of our examples, we use the Multiattribute Point notation when an *XR* unit defines a reserve price for only a single configuration.

We use a slightly more expressive bidding language in the multiattribute call markets implemented here, which is an *OR* extension of the *XR* unit.

**Definition 9 (Multiattribute *OXR* Bid)** *A* multiattribute *OXR* bid *is a set of multiattribute XR units, $\{XR_1, XR_2, \ldots, XR_M\}$, indicating 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. The multiattribute point $(x, p, q)$ expresses the valuation equivalent to an *OR* expression over $|q|$ atomic $(x, p)$ valuations:

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

The additional quantity designation in a multiattribute point provides compactness over the equivalent *OR* expression when valuations are linear in quantity.

The multiattribute *XR* unit with quantity $q$ defines the valuation equivalent to the following expression over atomic valuations:

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

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 (Engel, Wellman, and Lochner, 2006). The multiattribute *OXR* bid provides full expressiveness with respect to class *OXS*.

## Clearing

Previous work (Engel, Wellman, and Lochner, 2006) 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 presented above. The result holds for more general conditions on the bidding language as described in the earlier paper.

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 $XR_b = (configs^b, prices^b, q^b)$ and sell *XR* unit $XR_s = (configs^s, prices^s, q^s)$,

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

The *MMP surplus* is also necessary to compute a solution to *GMAP*:

$$MMP_s(XR_b, XR_s) = \max_{x \in X}[r(XR_b, x) - r(XR_s, x)].$$

Define the set *BX* as the set of all *XR* units contained in the buyers' *OXR* bids, and the set *SX* as the set of all *XR* units in the sellers' *OXR* bids. For $b_i = \{XR_{i,1}, XR_{i,2}, \ldots, XR_{i,M}\}$ for buyer $i \in C$ and $b_j = \{XR_{j,1}, XR_{j,2}, \ldots, XR_{j,M}\}$ for seller $j \in S$,

$$\begin{aligned} BX &= \bigcup_{i \in C} \bigcup_{XR_{i,k} \in b_i} XR_{i,k}, \\ SX &= \bigcup_{j \in S} \bigcup_{XR_{j,k} \in b_j} XR_{j,k}. \end{aligned}$$

In this case, the multiattribute matching problem is performed between each pair in $BX \times SX$. *GMAP* is then formulated as a network flow algorithm, 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.

## Information Feedback

In addition to reducing the complexity of clearing, the decomposition of *GMAP* into *MMP* and subsequent network optimization reduces the complexity of computing quote information. The single-unit quote defines a limit on the offer price to trade a single unit of a configuration. The *MMP* process translates this price into a bilateral surplus for each offer in the order book, and the network flow optimization determines whether the offer will transact based on these bilateral surpluses. A quote for a given configuration can be found by first finding the required surplus (i.e., solution to $MMP_s$) for a new bilateral trade to be included in the efficient set, and then determining the price level for any given configuration as a function of that 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.

As an example, consider calculating the bid quote for $x$ given a set of *XR* units comprised of buys *BX* and sells *SX*. Each *XR* unit is a node in the network flow graph, with link surplus defined as above. We first add a dummy node ($XR_D$) to the graph and connect that node to one of the nodes of *BX* ($XR_i$). We must now calculate the link surplus which increases the value of the optimal network flow. The computed link quote, $LQ_i$, is the trade surplus (i.e., solution to $MMP_s(XR_i, XR_D)$) which is required for a new bid to trade with node $XR_i$. The link quote for each buy node must be calculated, producing a link quote for each $XR_i \in BX$. The bid quote for a given configuration $x$ is then:

$$\max_{XR_i \in BX}(r(XR_i, x) - LQ_i).$$

It should be apparent from the max operation that once all link quotes have been determined, configuration quotes can be computed with complexity on the order of the number of *XR* units. This implies that the complexity of computing 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 that is 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.

The computation of link quotes on the network flow graph is also achievable in polynomial time, using a specialization of the cycle-canceling algorithm (Ahuja, Magnanti, and Orlin, 1993). 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 that is 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.

## Multiattribute Valuations

The call market presented above supports the direct expression of *OXS* valuations, with information feedback supporting the implementation of market-based algorithms for efficiency under *GS* valuations. Unfortunately, many valuations natural for multiattribute domains fall outside of class *OXS*. For example, Bichler and Kalagnanam (2005) cite the common need to enforce *homogeneity* of allocations, meaning that all configurations in an allocation must have identical values for one or more attributes. Valuations which place higher values on homogeneous allocations, or conversely, heterogeneous allocations, are expressible with an *XOS* bidding language but not with a language of class *OXS*.

Our motivation in studying allocation with valuations of increasing complexity arose when applying our mechanism to a supply chain manufacturing scenario. In the Trading Agent Competition Supply Chain Management Game (Arunachalam and Sadeh, 2005), manufacturers assemble finished goods from a limited inventory of available components. As Example 1 makes clear, the induced seller valuations fall outside of class *OXS*, and may violate the gross substitutes condition.

**Example 1** *Consider the case of a PC built from two components: cpu and memory. Assume that a seller has one unit of* $cpu = fast$*, one unit of* $cpu = slow$*, one unit each for* $memory \in \{large, medium, small\}$*, and the configurations are assembled in the following manner:*

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:*

$$\{x_1, x_4\}, \{x_1, x_3\}, \{x_2, x_3\}.$$

*The induced seller valuation is not expressible using an OXS language. The nearest OXR bid approximations require the seller to either overstate (bid 1) or understate (bids 2 and 3) his production capabilities:*

$$(((x_1, x_2), (p_1, p_2), -1), ((x_3, x_4), (p_3, p_4), -1)) \quad (1)$$

$$((x_1, p_1, -1), ((x_3, x_4), (p_3, p_4), -1)) \quad (2)$$

$$(((x_1, x_2), (p_1, p_2), -1), (x_3, p_3, -1)) \quad (3)$$

The valuations from Example 1 are also not in class *GS*. 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 equivalent *XOS* valuation would be:

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

Prices for configurations are also necessary to evaluate the gross substitutes condition. Assume the following configuration prices:

$$p(x_1) = 5;\ p(x_2) = 4;\ p(x_3) = 4;\ p(x_4) = 5.$$

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.

## A New Valuation Metric

Given the ability to implement market-based algorithms, the question remains as to the efficiency limitations of our market design when valuations are not contained in *GS*. To better characterize the settings for which market-based algorithms are appropriate, in this section we define a new metric that assesses the degree to which a valuation violates the *GS* conditions.

### Gross Substitutes Revisited

As defined above, the gross substitutes condition requires that the demand for goods be nondecreasing in the prices of other goods. The motivation behind this condition is that a price adjustment process will ultimately reach equilibrium if a price perturbation intended to reduce the demand of over-demanded goods does not reduce the demand for other under-demanded goods. Similarly, a price decrease intended to increase demand for under-demanded goods should not increase demand for over-demanded goods.

For valuation $v$ satisfying the gross substitutes condition, the demand correspondence condition holds for all price vectors and perturbations. Formally, given demand correspondence $d(v|\vec{p})$ as defined previously, the set of allocations maximizing the sum $v(g) - \sum_{x \in g} p_x$, for all vectors of configuration prices $\vec{p} = (p_{x_1}, p_{x_2}, \ldots, p_{x_n}) \in \Re^n_+$, and all single price perturbations $\overrightarrow{dp} \in \Re^n_+$, for any $g_1 \in d(v|\vec{p})$ there exists $g_2 \in d(v|\vec{q})$ such that $\{x \in g_1 | p_x = q_x\} \subset g_2$, where $\vec{q} = \vec{p} + \overrightarrow{dp}$.

### Gross Substitutes Violations

We hypothesize that the *degree* to which the gross substitutes condition is violated is a more accurate metric on valuations in assessing the likely existence of equilibrium and the efficiency of market-based algorithms.

For non-negative vectors $\vec{p}$ and $\overrightarrow{dp}$, again define $\vec{q} = \vec{p} + \overrightarrow{dp}$. For each $g_i \in d(v|\vec{p})$, we define the *gross substitutes violation* as follows:

$$GSV(v, \vec{p}, \vec{q}, g_i) = \min_{g \in d(v|\vec{q})} \mid \{x \in g_i | p_x = q_x\} \setminus \{x \in g | p_x = q_x\} \mid .$$

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

To simplify the exposition, we assume $d(v|\vec{p})$ maps to a single $g$ for any $\vec{p}$, and use $x \in d(v|\vec{p})$ to indicate a good from that demand set.

From any single price vector $\vec{p}$, define the gross substitutes violation of a valuation for that price vector as the average *GSV* over all minimal single-price perturbations which ensure a new demand set:

$$GSV(v, \vec{p}) =$$
$$\frac{1}{n}\sum_{i=1}^{n} \mid \{x_j \in d(v|\vec{p}) | p_x = q_x\} \setminus \{x \in d(v|\vec{q}) | p_x = q_x\} \mid .$$

where:

$$\vec{q} = (p_1, \ldots, p_i + dp_i, \ldots, p_n)$$

and

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

Next, define the *expected* gross substitutes violation for a valuation as the expected value of *GSV* for random $\vec{p}$:

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

where

$$\forall i \quad p_i \sim U[0, \bar{p}].$$

Our hypothesis is that for a set of bidders with valuations drawn from a small range of *EGSV* values, market-based algorithms will reach a similar degree of efficiency, where the realized efficiency is decreasing in the average *EGSV* value. This result would extend naturally to the case of gross substitutes (equivalently, an *EGSV* value of zero), where market-based algorithms achieve full efficiency. The intuition behind the *EGSV* metric, specifically for using the expected *GSV* of a valuation (the average, rather than the maximum or minimum) is that since the price trajectory of a market-based algorithm covers only a subset of the full price space, the average violation factors in the probability of seeing any specific violation.

## Testing the *EGSV*-Efficiency Relationship

To evaluate the relationship between *EGSV* values and market efficiency, we employed a component-based model of configurations like that presented in Example 1, where valuation complexity is determined by the constitution of the configurations, that is, the *configuration structure*, as well as by the respective inventory levels and component costs of sellers.

For example, a valuation defined on a configuration structure with three alternate configurations will violate *GS* to the extent that swapping production from one configuration to another requires additional components that are allocated to the third configuration. 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.

### Valuation Generation

Our experimental approach to generating a configuration structure is to generate random configurations until a total of $20$ unique configurations is produced. 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.

Once we have generated a set of 20 configurations, we randomly sample costs and inventory to generate a seller valuation. Sellers have both component inventories and costs drawn i.i.d. Seller inventory for each component is drawn from the discrete uniform distribution $[0, 3]$, while seller costs per component are drawn from the discrete uniform distribution $[30, 80]$.

We then test the induced valuation with respect to the same price distribution from which agent valuations are drawn. We sample prices, and for each price sample $\vec{p}$, we compute $GSV(v, \vec{p})$. To compute $GSV(v, \vec{p})$, we first compute the optimal production set $g^* = d(v|\vec{p})$, and sum the gross substitutes violations over all minimal single prices changes which ensure a new optimal production set.

We iterate the above process with random price samples until the standard error of the expected gross substitutes violation is below $.05$, producing a single valuation. 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 tested seller valuations for 277 configuration structures, yielding a total of 27700 seller valuations.

### Market Simulation

Each problem instance comprises a set of 10 buyers and 10 sellers. For each configuration structure, we first sort the set of 100 generated seller valuations by *EGSV* values. 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 of the 10 sellers to classify the problem instance (we denote this average *EGSV* by *aGSV*). 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 among the goods (i.e., each will accept any combination of two goods at probabilistically generated configuration reserve prices). Buyer reserve prices are drawn from the discrete uniform distribution $[400, 500]$ for each configuration.

For each problem instance, we first formulate the allocation problem as a linear program to determine the maximum achievable efficiency. We then simulate bidding until quiescence, computing the fraction of maximal efficiency at quiescence. To quantify the benefit of information feedback, we took the first iteration of bidding as the direct-revelation outcome.

Although we believe that the direct expression of substitutes—as in the *OXR* bidding language—is indispensable for large or continuous attribute domains, we conducted the same bidding simulation with a class *OS* bidding language to determine whether the expression of substitutes provides efficiency advantages in domains with small numbers of configurations. Each problem instance thus produces 4 data points (1 for each of $(direct, iterative) \times (OS, OXR)$).

We simulated myopic best response bidding for buyers and sellers, having buyers and sellers bid their true values at each iteration on a profit-maximizing set of goods. Given that the bidding language is not sufficiently expressive to directly reveal seller valuations, sellers are forced to approximate their valuations with bids. To generate an *OS* bid, sellers find the feasible production bundle that maximizes surplus at current prices (assuming a uniform price for all configurations when quotes are not available). To generate an optimal *OXR* bid, sellers start with the optimal *OS* bid, subsequently expanding the bid to a feasible *OXR* bid.

## Simulation Results

We aggregated the simulation results over all configuration structures and sorted the data by aGSV value into 10 bins, computing the sample mean of the achieved fraction of total efficiency. Figure 1 plots the achieved fraction of maximal efficiency as a function of aGSV value (with aGSV value averaged over all samples within a bin), for both direct-revelation and iterative mechanisms, for both the *OS* and *OXR* bidding languages.

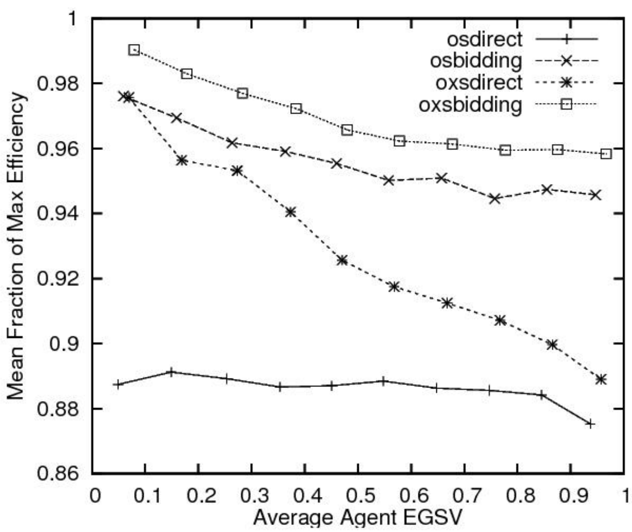


Figure 1: Mean efficiency for average realized *EGSV*.

For aGSV values close to zero, the substitutes condition is nearly satisfied for all valuations, indicating that prices inducing violations of the substitutes condition occur infrequently. For such problem instances, we would expect iterative mechanisms to reach nearly maximal efficiency. Figure 1 confirms this hypothesis, as both iterative mechanisms average more than $97\%$ efficiency for low aGSV values.

We note that while the direct *OS* mechanism suffers from relatively lower efficiency across all aGSV values ($\sim 90\%$), the direct *OXR* mechanism achieves a high fraction of efficiency for low aGSV values ($\sim 97\%$). We conjecture that the majority of low *EGSV* valuations were also in class *OXS*, and that given *OXS* valuations, the ability to express substitutability through bids provides a significant efficiency advantage in direct-revelation mechanisms.

Notable in Figure 1 is that the iterative mechanisms relatively outperform the direct *OXR* mechanism, by a margin that is increasing in aGSV value. We suspect this reflects bidder valuations which increasingly deviate from class *OXS* with higher *EGSV* values. Despite this increasing valuation complexity, the iterative mechanisms hold to a high level of efficiency, falling only to approximately $95\%$ as aGSV values reach $1$. From this we conclude that information feedback is able to compensate for the lack of expressive power of a class *OXS* bidding language.

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.

## Conclusions

We investigated the problem of multi-unit multiattribute allocation through call-market auctions. We presented an implemented multiattribute call market supporting polynomial-time clearing and information feedback operations for a restricted class of bidding language. To our knowledge, this is the first call market of its kind presented in literature.

We discussed the expected efficiency of our mechanism from the perspective of known hardness results derived for combinatorial auction settings, given complement-free bidder valuations. We discussed the efficiency of market-based algorithms, and demonstrated that the information-feedback functionality of our market design supports efficient allocations given valuations satisfying the gross substitutes condition.

Importantly, we demonstrated that the addition of information feedback functionality to our call market design successfully compensates for the expressive deficiencies imposed by a restricted bidding language. The addition of information feedback support to our previously developed clearing algorithm thus extends the range of bidder valuations for which our market design is able to support efficiency.

Inspired by a multiattribute supply chain setting, we presented natural ways in which multiattribute valuations may violate the gross substitutes condition. Lacking theoretical results as to the expected efficiency of our market design for valuations likely to be encountered in practice, we presented a new metric on bidder valuations, derived from the ways in which valuations violate the substitutes condition. We then presented evidence that this metric correlates with the expected efficiency of market-based algorithms. Also to our knowledge, this is the first such experiment presenting evidence of a correlation between the efficiency of market-based algorithms and the measured gross substitutes violation of bidder valuations.

# References


Ahuja, R. K.; Magnanti, T. L.; and Orlin, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc.

Arrow, K. J.; Block, H. D.; and Hurwicz, L. 1959. On the stability of competitive equilibrium, II. *Econometrica* 27:82–109.

Arunachalam, R., and Sadeh, N. M. 2005. The supply chain trading agent competition. *Electronic Commerce Research and Applications* 4:63–81.

Bichler, M., and Kalagnanam, J. 2005. Configurable offers and winner determination in multi-attribute auctions. *European Journal of Operational Research* 160:380–394.

Bichler, M.; Kaukal, M.; and Segev, A. 1999. Multi-attribute auctions for electronic procurement. In *First IBM IAC Workshop on Internet Based Negotiation Technologies*.

Branco, F. 1997. The design of multidimensional auctions. *RAND Journal of Economics* 28:63–81.

Che, Y.-K. 1997. Design competition through multidimensional auctions. *RAND Journal of Economics* 24:668–680.

Cramton, P.; Shoham, Y.; and Steinberg, R., eds. 2005. *Combinatorial Auctions*. MIT Press.

Dobzinski, S.; Nisan, N.; and Schapira, M. 2005. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Thirty-Seventh Annual ACM Symposium on Theory of Computing*, 610–618.

Engel, Y.; Wellman, M. P.; and Lochner, K. M. 2006. Bid expressiveness and clearing algorithms in multiattribute double auctions. In *Seventh ACM Conference on Electronic Commerce*, 110–119.

Gul, F., and Stacchetti, E. 1999. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory* 87(1):95–124.

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

Lehmann, B.; Lehmann, D.; and Nisan, N. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55(2):270–296.

Mas-Colell, A.; Whinston, M. D.; and Green, J. R. 1995. *Microeconomic Theory*. New York: Oxford University Press.

Milgrom, P., and Strulovici, B. 2006. Concepts and properties of substitute goods. Technical report, Economics Group, Nuffield College, University of Oxford.

Parkes, D. C., and Kalagnanam, J. 2005. Models for iterative multiattribute Vickrey auctions. *Management Science* 51:435–451.

Sandholm, T.; Suri, S.; Gilpin, A.; and Levine, D. 2002. Winner determination in combinatorial auction generalizations. In *First International Joint Conference on Autonomous Agents and Multiagent Systems*, 69–76.

Sandholm, T. 2005. The winner determination problem. In Cramton et al. Cramton, Shoham, and Steinberg (2005).

Segal, I. 2005. The communication requirements of combinatorial allocation problems. In Cramton et al. Cramton, Shoham, and Steinberg (2005).

Sunderam, A. V., and Parkes, D. C. 2003. Preference elicitation in proxied multiattribute auctions. In *Fourth ACM Conference on Electronic Commerce*, 214–215.

Wurman, P. R.; Wellman, M. P.; and Walsh, W. E. 2001. A parametrization of the auction design space. *Games and Economic Behavior* 35:304–338.