

Bid Expressiveness and Clearing Algorithms in Multiattribute Double Auctions

Yagil Engel, Michael P. Wellman, and Kevin M. Lochner

University of Michigan, Computer Science & Engineering
2260 Hayward St, Ann Arbor, MI 48109-2121, USA

{yagil,wellman,klochner}@umich.edu

ABSTRACT

We investigate the space of two-sided multiattribute auctions, focusing on the relationship between constraints on the offers traders can express through bids, and the resulting computational problem of determining an optimal set of trades. We develop a formal semantic framework for characterizing expressible offers, and show conditions under which the allocation problem can be separated into first identifying optimal pairwise trades and subsequently optimizing combinations of those trades. We analyze the bilateral matching problem while taking into consideration relevant results from multiattribute utility theory. Network flow models we develop for computing global allocations facilitate classification of the problem space by computational complexity, and provide guidance for developing solution algorithms. Experimental trials help distinguish tractable problem classes for proposed solution techniques.

Categories and Subject Descriptors: F.2 [Theory of Computation]: Analysis Of Algorithms And Problem Complexity; J.4 [Computer Applications]: Social and Behavioral Sciences—Economics

General Terms: Algorithms, Economics

Keywords: Auctions, Multiattribute Auctions

1. BACKGROUND

A *multiattribute auction* is a market-based mechanism where goods are described by vectors of features, or *attributes* [3, 5, 8, 18]. Such mechanisms provide traders with the ability to negotiate over a multidimensional space of potential deals, delaying commitment to specific configurations until the most promising candidates are identified. For example, in a multiattribute auction for computers, the good may be defined by attributes such as processor speed, memory, and hard disk capacity. Agents have varying preferences (or costs) associated with the possible configurations. For example, a buyer may be willing to purchase a computer with a 2 GHz processor, 500 MB of memory, and a 50 GB hard disk for a price no greater than \$500, or the same computer with 1GB of memory for a price no greater than \$600.

Existing research in multiattribute auctions has focused primarily on one-sided mechanisms, which automate the process whereby a single agent negotiates with multiple potential trading partners [8, 7, 18, 5, 22, 21]. Models of procurement typically assume the buyer has a value function, v , ranging over the possible configurations, X , and that each seller i can similarly be associated with a cost function c_i over this domain. The role of the auction is to elicit these functions (possibly approximate or partial versions), and identify the surplus-maximizing deal. In this case, such an outcome would be $\arg \max_{i,x} v(x) - c_i(x)$. This problem can be translated into the more familiar auction for a single good without attributes by computing a *score* for each attribute vector based on the seller valuation function, and have buyers bid scores. Analogs of the classic first- and second-price auctions correspond to first- and second-score auctions [8, 7].

In the absence of a published buyer scoring function, agents on both sides may provide partial specifications of the deals they are willing to engage. Research on such auctions has, for example, produced iterative mechanisms for eliciting cost functions incrementally [18]. Other efforts focus on the optimization problem facing the bid taker, for example considering *side constraints* on the combination of trades comprising an overall deal [4]. Side constraints have also been analyzed in the context of combinatorial auctions [6, 19].

Our emphasis is on *two-sided* multiattribute auctions, where multiple buyers and sellers submit bids, and the objective is to construct a set of deals maximizing overall surplus. Previous research on such auctions includes works by Fink et al. [11] and Gong [13], both of which consider a matching problem for *continuous* double auctions (CDAs), where deals are struck whenever a pair of compatible bids is identified.

In a *call market*, in contrast, bids accumulate until designated times (e.g., on a periodic or scheduled basis) at which the auction *clears* by determining a comprehensive match over the entire set of bids. Because the optimization is performed over an aggregated scope, call markets often enjoy liquidity and efficiency advantages over CDAs [10].¹

Clearing a multiattribute CDA is much like clearing a one-sided multiattribute auction. Because nothing happens between bids, the problem is to match a given new bid (say, an offer to buy) with the existing bids on the other (sell) side. Multiattribute call markets are potentially much more complex. Constructing an optimal overall matching may require consideration of many different combina-

¹In the interim between clears, call markets may also disseminate *price quotes* providing summary information about the state of the auction [23]. Such price quotes are often computed based on hypothetical clears, and so the clearing algorithm may be invoked more frequently than actual market clearing operations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC'06, June 11–15, 2006, Ann Arbor, Michigan, USA.
Copyright 2006 ACM 1-59593-236-4/06/0006 ...\$5.00.

tions of trades, among the various potential trading-partner pairings. The problem can be complicated by restrictions on overall assignments, as expressed in side constraints [15].

The goal of the present work is to develop a general framework for multiattribute call markets, to enable investigation of design issues and possibilities. In particular, we use the framework to explore tradeoffs between expressive power of agent bids and computational properties of auction clearing. We conduct our exploration independent of any consideration of strategic issues bearing on mechanism design. As with analogous studies of combinatorial auctions [17], we intend that tradeoffs quantified in this work can be combined with incentive factors within a comprehensive overall approach to multiattribute auction design.

We provide the formal semantics of multiattribute offers in our framework in the next section. We abstract, where appropriate, from the specific language used to express offers, characterizing expressiveness semantically in terms of what deals may be offered. This enables us to identify some general conditions under which the problem of multilateral matching can be decomposed into bilateral matching problems. We then develop a family of network flow problems that capture corresponding classes of multiattribute call market optimizations. Experimental trials provide preliminary confirmation that the network formulations provide useful structure for implementing clearing algorithms.

2. MULTIATTRIBUTE OFFERS

2.1 Basic Definitions

The distinguishing feature of a multiattribute auction is that the goods are defined by vectors of *attributes*, $x = (x_1, \dots, x_m)$, $x_j \in X_j$. A *configuration* is a particular attribute vector, $x \in X = \prod_{j=1}^m X_j$. The outcome of the auction is a *set of bilateral trades*. Trade t takes the form $t = (x, q, b, s, \pi)$, signifying that agent b buys $q > 0$ units of configuration x from seller s , for payment $\pi > 0$. For convenience, we use the notation x_t to denote the configuration associated with trade t (and similarly for other elements of t). For a set of trades T , we denote by T_i that subset of T involving agent i (i.e., $b = i$ or $s = i$). Let \mathcal{T} denote the set of all possible trades.

A *bid* expresses an agent's willingness to participate in trades. We specify the semantics of a bid in terms of *offer sets*. Let $\mathcal{O}_i^T \subseteq \mathcal{T}_i$ denote agent i 's *trade offer set*. Intuitively, this represents the trades in which i is willing to participate. However, since the outcome of the auction is a *set* of trades, several of which may involve agent i , we must in general consider willingness to engage in *trade combinations*. Accordingly, we introduce the *combination offer set* of agent i , $\mathcal{O}_i^C \subseteq 2^{\mathcal{T}_i}$.

2.2 Specifying Offer Sets

A fully expressive bid language would allow specification of arbitrary combination offer sets. We instead consider a more limited class which, while restrictive, still captures most forms of multiattribute bidding proposed in the literature. Our bids directly specify part of the agent's trade offer set, and include further directives controlling how this can be extended to the full trade and combination offer sets.

For example, one way to specify a trade (buy) offer set would be to describe a set of configurations and quantities, along with the maximal payment one would exchange for each (x, q) specified. This description could be by enumeration, or any available means of defining such a mapping.

An explicit set of trades in the offer set generally entails inclusion of many more implicit trades. We assume *payment monotonicity*,

which means that agents always prefer more money. That is, for $\pi > \pi' > 0$,

$$\begin{aligned} (x, q, i, s, \pi) \in \mathcal{O}_i^T &\Rightarrow (x, q, i, s, \pi') \in \mathcal{O}_i^T, \\ (x, q, b, i, \pi') \in \mathcal{O}_i^T &\Rightarrow (x, q, b, i, \pi) \in \mathcal{O}_i^T. \end{aligned}$$

We also assume *free disposal*, which dictates that for all $i, q > q' > 0$,

$$\begin{aligned} (x, q', i, s, \pi) \in \mathcal{O}_i^T &\Rightarrow (x, q, i, s, \pi) \in \mathcal{O}_i^T, \\ (x, q, b, i, \pi) \in \mathcal{O}_i^T &\Rightarrow (x, q', b, i, \pi) \in \mathcal{O}_i^T. \end{aligned}$$

Note that the conditions for agents in the role of buyers and sellers are analogous. Henceforth, for expository simplicity, we present all definitions with respect to buyers only, leaving the definition for sellers as understood. Allowing agents' bids to comprise offers from both buyer and seller perspectives is also straightforward.

An assertion that offers are *divisible* entails further implicit members in the trade offer set.

DEFINITION 1 (DIVISIBLE OFFER). *Agent i 's offer is divisible down to q iff*

$$\forall q' < q. (x, q, i, s, \pi) \in \mathcal{O}_i^T \Rightarrow (x, q', i, s, \frac{q'}{q}\pi) \in \mathcal{O}_i^T.$$

We employ the shorthand divisible to mean divisible down to 0.

The definition above specifies arbitrary divisibility. It would likewise be possible to define divisibility with respect to integers, or to any given finite granularity. Note that when offers are divisible, it suffices to specify one offer corresponding to the maximal quantity one is willing to trade for any given configuration, trading partner, and per-unit payment (called the *price*).

At the extreme of indivisibility are *all-or-none* offers.

DEFINITION 2 (AON OFFER). *Agent i 's offer is all-or-none (AON) iff*

$$(x, q, i, s, \pi) \in \mathcal{O}_i^T \wedge (x, q', i, s, \pi') \in \mathcal{O}_i^T \Rightarrow [q = q' \vee \pi = \pi'].$$

In many cases, the agent will be indifferent with respect to different trading partners. In that event, it may omit the partner element from trades directly specified in its offer set, and simply assert that its offer is *anonymous*.

DEFINITION 3 (ANONYMITY). *Agent i 's offer is anonymous iff $\forall s, s', b, b'. (x, q, i, s, \pi) \in \mathcal{O}_i^T \Leftrightarrow (x, q, i, s', \pi) \in \mathcal{O}_i^T \wedge (x, q, b, i, \pi) \in \mathcal{O}_i^T \Leftrightarrow (x, q, b', i, \pi) \in \mathcal{O}_i^T$.*

Because omitting trading partner qualifications simplifies the exposition, we generally assume in the following that all offers are anonymous unless explicitly specified otherwise. Extending to the non-anonymous case is conceptually straightforward. We employ the wild-card symbol $*$ in place of an agent identifier to indicate that any agent is acceptable.

To specify a trade offer set, a bidder directly specifies a set of willing trades, along with any regularity conditions (e.g., divisibility, anonymity) that implicitly extend the set. The full trade offer set is then defined by the closure of this direct set with respect to payment monotonicity, free disposal, and any applicable divisibility assumptions.

We next consider the specification of combination offer sets. Without loss of generality, we restrict each trade set $T \in \mathcal{O}_i^C$ to include at most one trade for any combination of configuration and trading partner (multiple such trades are equivalent to one net trade aggregating the quantities and payments). The key question is to what extent the agent is willing to aggregate deals *across* configurations or trading partners. One possibility is disallowing any aggregation.

DEFINITION 4 (NO AGGREGATION). *The no-aggregation combinations are given by $\mathcal{O}_i^{NA} = \{\emptyset\} \cup \{\{t\} \mid t \in \mathcal{O}_i^T\}$. Agent i 's offer exhibits non-aggregation iff $\mathcal{O}_i^C = \mathcal{O}_i^{NA}$.*

We require in general that $\mathcal{O}_i^C \supseteq \mathcal{O}_i^{NA}$.

A more flexible policy is to allow aggregation across trading partners, keeping configuration constant.

DEFINITION 5 (PARTNER AGGREGATION). *Suppose a particular trade is offered in the same context (set of additional trades, T) with two different sellers, s and s' . That is,*

$$\{(x, q, i, s, \pi)\} \cup T \in \mathcal{O}_i^C \wedge \{(x, q, i, s', \pi)\} \cup T \in \mathcal{O}_i^C.$$

Agent i 's offer allows seller aggregation iff in all such cases,

$$\{(x, q', i, s, \pi'), (x, q - q', i, s', \pi - \pi')\} \cup T \in \mathcal{O}_i^C.$$

In other words, we may create new trade offer combinations by splitting the common trade (quantity and payment, not necessarily proportionately) between the two sellers.

In some cases, it might be reasonable to form combinations by aggregating *different* configurations.

DEFINITION 6 (CONFIGURATION AGGREGATION). *Suppose agent i offers, in the same context, the same quantity of two (not necessarily different) configurations, x and x' . That is,*

$$\{(x, q, i, *, \pi)\} \cup T \in \mathcal{O}_i^C \wedge \{(x', q, i, *, \pi')\} \cup T \in \mathcal{O}_i^C.$$

Agent i 's offer allows configuration aggregation iff in all such cases (and analogously when it is a seller),

$$\{(x, q', i, *, \frac{q'}{q}\pi), (x', q - q', i, *, \frac{q - q'}{q}\pi')\} \cup T \in \mathcal{O}_i^C.$$

Note that combination offer sets can accommodate offerings of configuration bundles. However, classes of bundles formed by partner or configuration aggregation are highly regular, covering only a specific type of bundle formed by splitting a desired quantity across configurations. This is quite restrictive compared to the general combinatorial case.

2.3 Willingness to Pay

An agent's offer trade set implicitly defines the agent's *willingness to pay* for any given configuration and quantity. We assume anonymity to avoid conditioning our definitions on trading partner.

DEFINITION 7 (WILLINGNESS TO PAY). *Agent i 's willingness to pay for quantity q of configuration x is given by*

$$\hat{u}_i^B(x, q) = \max \pi \text{ s.t. } (x, q, i, *, \pi) \in \mathcal{O}_i^T.$$

We use the symbol \hat{u} to recognize that willingness to pay can be viewed as a proxy for the agent's utility function, measured in monetary units. The superscript B distinguishes the buyer's willingness-to-pay function from, a seller's *willingness to accept*, $\hat{u}_i^S(x, q)$, defined as the *minimum* payment seller i will *accept* for q units of configuration x . We omit the superscript where the distinction is inessential or clear from context.

DEFINITION 8 (TRADE QUANTITY BOUNDS). *Agent i 's minimum trade quantity for configuration x is given by*

$$q_i(x) = \min q \text{ s.t. } \exists \pi. (x, q, i, *, \pi) \in \mathcal{O}_i^T.$$

The agent's maximum trade quantity for x is

$$\bar{q}_i(x) = \max q \text{ s.t.}$$

$$\exists \pi. (x, q, i, *, \pi) \in \mathcal{O}_i^T \wedge \neg \exists q' < q. (x, q', i, *, \pi) \in \mathcal{O}_i^T.$$

When the agent has no offers involving x , we take $q_i(x) = \bar{q}_i(x) = 0$.

It is useful to define a special case where all configurations are offered in the same quantity range.

DEFINITION 9 (CONFIGURATION PARITY). *Agent i 's offers exhibit configuration parity iff*

$$q_i(x) > 0 \wedge q_i(x') > 0 \Rightarrow q_i(x) = q_i(x') \wedge \bar{q}_i(x) = \bar{q}_i(x').$$

Under configuration parity we drop the arguments from trade quantity bounds, yielding the constants \bar{q} and q which apply to all offers.

DEFINITION 10 (LINEAR PRICING). *Agent i 's offers exhibit linear pricing iff for all $q_i(x) \leq q \leq \bar{q}_i(x)$,*

$$\hat{u}_i(x, q) = \frac{q}{\bar{q}_i(x)} \hat{u}_i(x, \bar{q}_i(x)).$$

Note that linear pricing assumes divisibility down to $q_i(x)$. Given linear pricing, we can define the *unit willingness to pay*, $\hat{u}_i(x) = \hat{u}_i(x, \bar{q}_i(x)) / \bar{q}_i(x)$, and take $\hat{u}_i(x, q) = q \hat{u}_i(x)$ for all $q_i(x) \leq q \leq \bar{q}_i(x)$.

In general, an agent's willingness to pay may depend on a context of other trades the agent is engaging in.

DEFINITION 11 (WILLINGNESS TO PAY IN CONTEXT). *Agent i 's willingness to pay for quantity q of configuration x in the context of other trades T is given by*

$$\hat{u}_i^B(x, q; T) = \max \pi \text{ s.t. } \{(x, q, i, s, \pi)\} \cup T_i \in \mathcal{O}_i^C.$$

LEMMA 1. *If \mathcal{O}_i^C is either non aggregating, or exhibits linear pricing, then*

$$\hat{u}_i^B(x, q; T) = \hat{u}_i^B(x, q).$$

3. MULTIATTRIBUTE ALLOCATION

DEFINITION 12 (TRADE SURPLUS). *The surplus of trade $t = (x, q, b, s, \pi)$ is given by*

$$\sigma(t) = \hat{u}_b^B(x, q) - \hat{u}_s^S(x, q).$$

Note that the trade surplus does not depend on the payment, which is simply a transfer from buyer to seller.

DEFINITION 13 (TRADE UNIT SURPLUS). *The unit surplus of trade $t = (x, q, b, s, \pi)$ is given by $\sigma^1(t) = \sigma(t)/q$.*

Under linear pricing, we can equivalently write $\sigma^1(t) = \hat{u}_b^B(x) - \hat{u}_s^S(x)$.

DEFINITION 14 (SURPLUS OF A TRADE IN CONTEXT). *The surplus of trade $t = (x, q, b, s, \pi)$ in the context of other trades T , $\sigma(t; T)$, is given by*

$$\hat{u}_b^B(x, q; T) - \hat{u}_s^S(x, q; T).$$

DEFINITION 15 (GMAP). *The Global Multiattribute Allocation Problem (GMAP) is to find the set of acceptable trades maximizing total surplus,*

$$\max_{T \in 2^T} \sum_{t \in T} \sigma(t; T \setminus \{t\}) \text{ s.t. } \forall i. T_i \in \mathcal{O}_i^C.$$

DEFINITION 16 (MMP). *The Multiattribute Matching Problem (MMP) is to find a best trade for a given pair of traders,*

$$\text{MMP}(b, s) = \arg \max_{t \in \mathcal{O}_b^T \cap \mathcal{O}_s^T} \sigma(t).$$

If $\mathcal{O}_b^T \cap \mathcal{O}_s^T$ is empty, we say that MMP has no solution.

Proofs of all the following results are provided in an extended version of this paper available from the authors.

THEOREM 2. *Suppose all agents' offers exhibit no aggregation (Definition 4). Then the solution to GMAP consists of a set of trades, each of which is a solution to MMP for its specified pair of traders.*

THEOREM 3. *Suppose that each agent's offer set satisfies one of the following (not necessarily the same) sets of conditions.*

1. *No aggregation and configuration parity (Definitions 4 and 9).*
2. *Divisibility, linear pricing, and configuration parity (Definitions 1, 10, and 9), with combination offer set defined as the minimal set consistent with configuration aggregation (Definition 6).²*

Then the solution to GMAP consists of a set of trades, each of which employs a configuration that solves MMP for its specified pair of traders.

Let $MMP^d(b, s)$ denote a modified version of MMP , where \mathcal{O}_b^T and \mathcal{O}_s^T are extended to assume divisibility (i.e., the offer sets are taken to be their closures under Definition 1). Then we can extend Theorem 3 to allow aggregating agents to maintain AON or min-quantity offers as follows.

THEOREM 4. *Suppose offer sets as in Theorem 3, except that agents i satisfying configuration aggregation need be divisible only down to q_i , rather than down to 0. Then the solution to GMAP consists of a set of trades, each of which employs the same configuration as a solution to MMP^d for its specified pair of traders.*

THEOREM 5. *Suppose agents b and s exhibit configuration parity, divisibility, and linear pricing, and there exists configuration x such that $\hat{u}_b(x) - \hat{u}_s(x) > 0$. Then $t \in MMP^d(b, s)$ iff*

$$\begin{aligned} x_t &= \arg \max_x \{ \hat{u}_b(x) - \hat{u}_s(x) \} \\ q_t &= \min(\bar{q}_b, \bar{q}_s). \end{aligned} \quad (1)$$

The preceding results signify that under certain conditions, we can divide the global optimization problem into two parts: first find a bilateral trade that maximizes unit surplus for each pair of traders (or total surplus in the non-aggregation case), and then use the results to find a globally optimal set of trades. In the following two sections we investigate each of these subproblems.

4. UTILITY REPRESENTATION AND MMP

We turn next to consider the problem of finding a best deal between pairs of traders. The complexity of MMP depends pivotally on the representation by bids of offer sets, an issue we have postponed to this point.

Note that issues of utility representation and MMP apply to a broad class of multiattribute mechanisms, beyond the multiattribute call markets we emphasize. For example, the complexity results contained in this section apply equally to the bidding problem faced by sellers in reverse auctions, given a published buyer scoring function.

The simplest representation of an offer set is a direct enumeration of configurations and associated quantities and payments. This approach treats the configurations as atomic entities, making no use

²That is, for such an agent i , \mathcal{O}_i^C is the closure under configuration aggregation of \mathcal{O}_i^{NA} .

of attribute structure. A common and inexpensive enhancement is to enable a trader to express sets of configurations, by specifying subsets of the domains of component attributes. Associating a single quantity and payment with a set of configurations expresses indifference among them; hence we refer to such a set as an *indifference range*.³ Indifference ranges include the case of attributes with a natural ordering, in which a bid specifies a minimum or maximum acceptable attribute level. The use of indifference ranges can be convenient for MMP . The compatibility of two indifference ranges is simply found by testing set intersection for each attribute, as demonstrated by the decision-tree algorithm of Fink et al. [11].

Alternatively, bidders may specify willingness-to-pay functions \hat{u} in terms of compact functional forms. Enumeration based representations, even when enhanced with indifference ranges, are ultimately limited by the exponential size of attribute space. Functional forms may avoid this explosion, but only if \hat{u} reflects structure among the attributes. Moreover, even given a compact specification of \hat{u} , we gain computational benefits only if we can perform the matching without expanding the \hat{u} values of an exponential number of configuration points.

4.1 Additive Forms

One particularly useful multiattribute representation is known as the additive scoring function. Though this form is widely used in practice and in the academic literature, it is important to stress the assumptions behind it. The theory of multiattribute representation is best developed in the context where \hat{u} is interpreted as a *utility function* representing an underlying preference order [16]. We present the premises of additive utility theory in this section, and discuss some generalizations in the next.

DEFINITION 17. *A set of attributes $Y \subset X$ is preferentially independent (PI) of its complement $Z = X \setminus Y$ if the conditional preference order over Y given a fixed level Z^0 of Z is the same regardless of the choice of Z^0 .*

In other words, the preference order over the projection of X on the attributes in Y is the same for any instantiation of the attributes in Z .

DEFINITION 18. *$X = \{x_1, \dots, x_m\}$ is mutually preferentially independent (MPI) if any subset of X is preferentially independent of its complement.*

THEOREM 6 ([9]). *A preference order over set of attributes X has an additive utility function representation*

$$u(x_1, \dots, x_m) = \sum_{i=1}^m u_i(x_i)$$

iff X is mutually preferential independent.

A utility function over outcomes including money is *quasi-linear* if the function can be represented as a function over non-monetary attributes plus payments, π . Interpreting \hat{u} as a utility function over non-monetary attributes is tantamount to assuming quasi-linearity. Even when quasi-linearity is assumed, however, MPI over non-monetary attributes is not sufficient for the quasi-linear utility function to be additive. For this, we also need that each of the pairs (π, X_i) for any attribute X_i would be PI of the rest of the attributes.

³These should not be mistaken with *indifference curves*, which express dependency between the attributes. Indifference curves can be expressed by the more elaborate utility representations discussed below.

This (by MAUT) in turn implies that the set of attributes including money is MPI and the utility function can be represented as

$$u(x_1, \dots, x_m, \pi) = \sum_{i=1}^m u_i(x_i) + \pi.$$

Given that form, a willingness-to-pay function reflecting u can be represented additively, as

$$\hat{u}(x) = \sum_{i=1}^m u_i(x_i)$$

In many cases the additivity assumption provides practically crucial simplification of offer set elicitation. In addition to compactness, additivity dramatically simplifies *MMP*. If both sides provide additive \hat{u} representations, the globally optimal match reduces to finding the optimal match separately for each attribute.

A common scenario in procurement has the buyer define an additive scoring function, while suppliers submit enumerated offer points or indifference ranges. This model is still very amenable to *MMP*: for each element in a supplier's enumerated set, we optimize each attribute by finding the point in the supplier's allowable range that is most preferred by the buyer.

A special type of scoring (more particularly, cost) function was defined by Bichler and Kalagnanam [4] and called a *configurable offer*. This idea is geared towards procurement auctions: assuming suppliers are usually comfortable with expressing their preferences in terms of cost that is quasi-linear in every attribute, they can specify a price for a base offer, and additional cost for every change in a specific attribute level. This model is essentially a "pricing out" approach [16]. For this case, *MMP* can still be optimized on a per-attribute basis. A similar idea has been applied to one-sided iterative mechanisms [18], in which sellers refine prices on a per-attribute basis at each iteration.

4.2 Multiattribute Utility Theory

Under MPI, the tradeoffs between the attributes in each subset cannot be affected by the value of other attributes. For example, when buying a PC, a weaker CPU may increase the importance of the RAM compared to, say, the type of keyboard. Such relationships cannot be expressed under an additive model.

Multiattribute utility theory (MAUT) develops various compact representations of utility functions that are based on weaker structural assumptions [16, 2]. There are several challenges in adapting these techniques to multiattribute bidding. First, as noted above, the theory is developed for utility functions, which may behave differently from willingness-to-pay functions. Second, computational efficiency of matching has not been an explicit goal of most work in the area. Third, adapting such representations to iterative mechanisms may be more challenging.

One representation that employs somewhat weaker assumptions than additivity, yet retains the summation structure is the *generalized additive (GA) decomposition*:

$$u(x) = \sum_{j=1}^J f_j(x^j), \quad x^j \in X^j, \quad (2)$$

where the X^j are potentially overlapping sets of attributes, together exhausting the space X .

A key point from our perspective is that the complexity of the matching is similar to the complexity of optimizing a single function, since the sum function is in the form (2) as well. Recent work by Gonzales and Perny [14] provides an elicitation process for GA

decomposable preferences under certainty, as well as an optimization algorithm for the GA decomposed function. The complexity of exact optimization is exponential in the induced width of the graph. However, to become operational for multiattribute bidding this decomposition must be detectable and verifiable by statements over preferences with respect to price outcomes. We are exploring this topic in ongoing work.

5. SOLVING *GMAP* UNDER ALLOCATION CONSTRAINTS

Theorems 2, 3, and 4 establish conditions under which *GMAP* solutions must comprise elements from constituent *MMP* solutions. In Sections 5.1 and 5.2, we show how to compute these *GMAP* solutions, given the *MMP* solutions, under these conditions. In these settings, traders that aggregate partners also aggregate configurations; hence we refer to them simply as "aggregating" or "non-aggregating". Section 5.3 suggests a means to relax the linear pricing restriction employed in these constructions. Section 5.4 provides strategies for allowing traders to aggregate partners and restrict configuration aggregation at the same time.

5.1 Notation and Graphical Representation

Our clearing algorithms are based on network flow formulations of the underlying optimization problem [1]. The network model is based on a bipartite graph, in which nodes on the left side represent buyers, and nodes on the right represent sellers. We denote the sets of buyers and sellers by B and S , respectively.

We define two graph families, one for the case of non-aggregating traders (called *single-unit*), and the other for the case of aggregating traders (called *multi-unit*).⁴ For both types, a single directed arc is placed from a buyer $i \in B$ to a seller $j \in S$ if and only if $MMP(i, j)$ is nonempty. We denote by $T(i)$ the set of potential trading partners of trader i (i.e., the nodes connected to buyer or seller i in the bipartite graph).

In the single-unit case, we define the weight of an arc (i, j) as $w_{i,j} = \sigma(MMP(i, j))$. Note that free disposal lets a buy offer receive a larger quantity than desired (and similarly for sell offers). For the multi-unit case, the weights are $w_{i,j} = \sigma^1(MMP(i, j))$, and we associate the quantity \bar{q}_i with the node for trader i . We also use the notation $q_{i,j}$ for the mathematical formulations to denote partial fulfillment of q_t for $t = MMP(i, j)$.

5.2 Handling Indivisibility and Aggregation Constraints

Under the restrictions of Theorems 2, 3, or 4, and when the solution to *MMP* is given, *GMAP* exhibits strong similarity to the problem of clearing double auctions with assignment constraints [15]. A match in our bipartite representation corresponds to a potential trade in which assignment constraints are satisfied. Network flow formulations have been shown to model this problem under the assumption of indivisibility and aggregation for all traders. The novelty in this part of our work is the use of generalized network flow formulations for more complex cases where aggregation and divisibility may be controlled by traders.

Initially we examine the simple case of no aggregation (Theorem 2). Observe that the optimal allocation is simply the solution to the well known *weighted assignment problem* [1] on the single-unit bipartite graph described above. The set of matches that maximizes the total weight of arcs corresponds to the set of trades that maximizes total surplus. Note that any form of (in)divisibility can

⁴In the next section, we introduce a hybrid form of graph accommodating mixes of the two trader categories.

also be accommodated in this model via the constituent *MMP* sub-problems.

The next formulation solves the case in which all traders fall under case 2 of Theorem 3—that is, all traders are aggregating and divisible, and exhibit linear pricing. This case can be represented using the following linear program, corresponding to our multi-unit graph:

$$\begin{aligned} \max \quad & \sum_{i \in B, j \in S} w_{ij} q_{ij} \\ \text{s.t.} \quad & \sum_{i \in T(j)} q_{ij} \leq \bar{q}_j \quad j \in S \\ & \sum_{j \in T(i)} q_{ij} \leq \bar{q}_i \quad i \in B \\ & q_{ij} \geq 0 \quad j \in S, i \in B \end{aligned}$$

Recall that the q_{ij} variables in the solution represent the number of units that buyer i procures from seller j . This formulation is known as the *network transportation problem with inequality constraints*, for which efficient algorithms are available [1]. It is a well known property of the transportation problem (and flow problems on pure networks in general) that given integer input values, the optimal solution is guaranteed to be integer as well. Figure 1 demonstrates the transformation of a set of bids to a transportation problem instance.

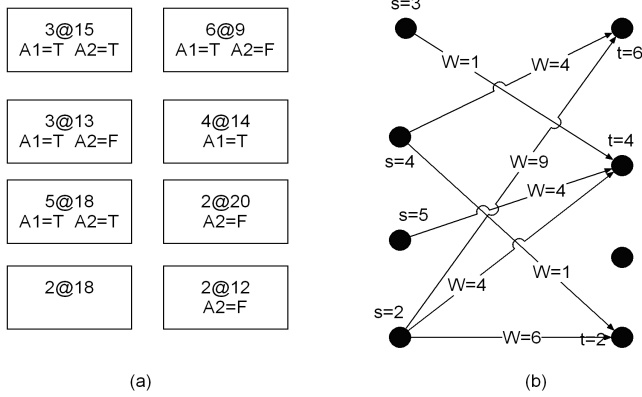


Figure 1: Multi-unit matching with two boolean attributes. (a) Bids, with offers to buy in the left column and offers to sell at right. $q@p$ indicates an offer to trade q units at price p per unit. Configurations are described in terms of constraints on attribute values. (b) Corresponding multi-unit assignment model. W represents arc weights (unit surplus), s represents source (exogenous) flow, and t represents sink quantity.

The problem becomes significantly harder when aggregation is given as an option to bidders, requiring various enhancements to the basic multi-unit bipartite graph described above. In general, we consider traders that are either aggregating or not, with either divisible or AON offers.

Initially we examine a special case, which at the same time demonstrates the hardness of the problem but still carries computational advantages. We designate one side (e.g., buyers) as restrictive (AON and non-aggregating), and the other side (sellers) as unrestrictive (divisible and aggregating). This problem can be represented using

the following integer programming formulation:

$$\begin{aligned} \max \quad & \sum_{i \in B, j \in S} w_{ij} q_{ij} \\ \text{s.t.} \quad & \sum_{i \in T(j)} \bar{q}_i q_{ij} \leq \bar{q}_j \quad j \in S \\ & \sum_{j \in T(i)} q_{ij} \leq 1 \quad i \in B \\ & q_{ij} \in \{0, 1\} \quad j \in S, i \in B \end{aligned} \quad (3)$$

This formulation is a restriction of the *generalized assignment problem (GAP)* [12]. Although GAP is known to be NP-hard, it can be solved relatively efficiently by exact or approximate algorithms. GAP is more general than the formulation above as it allows buy-side quantities (\bar{q}_i above) to be different for each potential seller. That this formulation is NP-hard as well (even the case of a single seller corresponds to the knapsack problem), illustrates the drastic increase in complexity when traders with different constraints are admitted to the same problem instance.

Other than the special case above, we found no advantage in limiting AON constraints when traders may specify aggregation constraints. Therefore, the next generalization allows any combination of the two boolean constraints, that is, any trader chooses among four bid types:

NI Bid AON and not aggregating.

AD Bid allows aggregation and divisibility.

AI Bid AON, allows aggregation (quantity can be aggregated across configurations, as long as it sums to the whole amount).

ND No aggregation, divisibility (one trade, but smaller quantities are acceptable).

To formulate an integer programming representation for the problem, we introduce the following variables. Boolean (0/1) variables r_i and r'_j indicate whether buyer i and seller j participate in the solution (used for AON traders). Another indicator variable, y_{ij} , applied to non-aggregating buyer i and seller j , is one iff i trades with j . For aggregating traders, y_{ij} is not constrained.

$$\max \quad \sum_{i \in B, j \in S} W_{ij} q_{ij} \quad (4a)$$

$$\text{s.t.} \quad \sum_{j \in T(i)} q_{ij} = \bar{q}_i r_i \quad i \in AI_b \quad (4b)$$

$$\sum_{j \in T(i)} q_{ij} \leq \bar{q}_i r_i \quad i \in AD_b \quad (4c)$$

$$\sum_{i \in T(j)} q_{ij} = \bar{q}_j r'_j \quad j \in AI_s \quad (4d)$$

$$\sum_{i \in T(j)} q_{ij} \leq \bar{q}_j r'_j \quad j \in AD_s \quad (4e)$$

$$x_{ij} \leq \bar{q}_i y_{ij} \quad i \in ND_b, j \in T(i) \quad (4f)$$

$$x_{ij} \leq \bar{q}_j y_{ij} \quad j \in NI_s, i \in T(j) \quad (4g)$$

$$\sum_{j \in T(i)} y_{ij} \leq r_i \quad i \in NI_b \cup ND_b \quad (4h)$$

$$\sum_{i \in T(j)} y_{ij} \leq r'_j \quad j \in NI_s \cup ND_s \quad (4i)$$

$$\text{int } q_{ij} \quad (4j)$$

$$y_{ij}, r'_j, r_i \in \{0, 1\} \quad (4k)$$

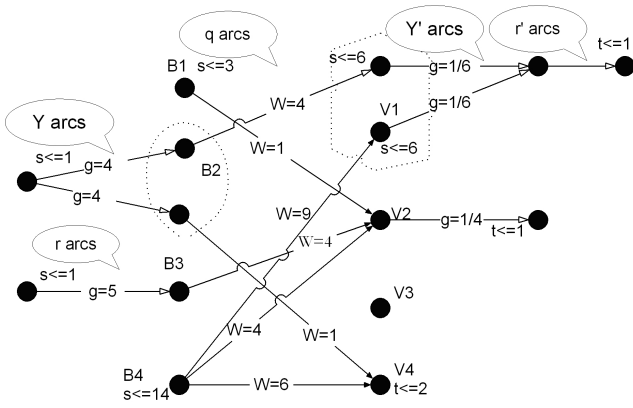


Figure 2: Generalized network flow model. $B1$ is a buyer in AD , $B2 \in NI$, $B3 \in AI$, $B4 \in ND$. $V1$ is a seller in ND , $V2 \in AI$, $V4 \in AD$. The g values represent arc gains.

Problem (4) has additional structure as a *generalized min-cost flow* problem with integral flow.⁵ A generalized flow network is a network in which each arc may have a *gain factor*, in addition to the pure network parameters (which are flow limits and costs). Flow in an arc is then multiplied by its gain factor, so that the flow that enters the end node of an arc equals the flow that entered from its start node, multiplied by the gain factor of the arc. The network model can in turn be translated into an IP formulation that captures such structure.

The generalized min-cost flow problem is well-studied and has a multitude of efficient algorithms [1]. The faster algorithms are polynomial in the number of arcs and the logarithm of the maximal gain, that is, performance is not strongly polynomial but is polynomial in the size of the input. The main benefit of this graphical formulation to our matching problem is that it provides a very efficient linear relaxation. Integer programming algorithms such as branch-and-bound use solutions to the linear relaxation instance to bound the optimal integer solution. Since network flow algorithms are much faster than arbitrary linear programs (generalized network flow simplex algorithms have been shown to run in practice only 2 or 3 times slower than pure network min-cost flow [1]), we expect a branch-and-bound solver for the matching problem to show improved performance when taking advantage of network flow modeling.

The network flow formulation is depicted in Figure 2. Non-restrictive traders are treated as in Figure 1. For a non-aggregating buyer, a single unit from the source will saturate up to one of the y_{i_j} for all j , and be multiplied by \bar{q}_i . If $i \in ND$, the end node of y_{i_j} will function as a sink that may drain up to \bar{q}_i of the entering flow. For $i \in NI$ we use an indicator (0/1) arc r_i , on which the flow is multiplied by \bar{q}_i . Trader i trades the full quantity iff $r_i = 1$.

At the seller side, the end node of a q_{i_j} arc functions as a source for sellers $j \in ND$, in order to let the flow through y'_{i_j} arcs be 0 or \bar{q}_j . The flow is then multiplied by $\frac{1}{\bar{q}_j}$ so 0/1 flows enter an end node which can drain either 1 or 0 units. For sellers $j \in NI$ arcs r'_j ensure AON similarly to arcs r_j for buyers.

Having established this framework, we are ready to accommo-

⁵Constraint (4j) could be omitted (yielding computational savings) if non-integer quantities are allowed. Here and henceforth we assume the harder problem, where divisibility is with respect to integers.

date more flexible versions of side constraints. The first generalization is to replace the boolean AON constraint with divisibility down to \bar{q} , the minimal quantity. In our network flow instance we simply need to turn the node of the constrained trader i (e.g., the node $B3$ in Figure 2) to a sink that can drain up to $\bar{q}_i - q_i$ units of flow. The integer program (4) can be also easily changed to accommodate this extension.

Using gains, we can also apply batch size constraints. If a trader specifies a batch size β , we change the gain on the r arcs to β , and set the available flow of its origin to the maximal number of batches \bar{q}_i/β .

5.3 Nonlinear Pricing

A key assumption in handling aggregation up to this point is linear pricing, which enables us to limit attention to a single unit price. Divisibility without linear pricing allows expression of concave willingness-to-pay functions, corresponding to convex preference relations. Bidders may often wish to express non-convex offer sets, for example, due to fixed costs or switching costs in production settings [20].

We consider nonlinear pricing in the form of enumerated payment *schedules*—that is, defining values $\hat{u}(x, q)$ for a select set of quantities q . For the indivisible case, these points are distinguished in the offer set by satisfying the following:

$$\exists \pi. (x, q, i, *, \pi) \in \mathcal{O}_i^T \wedge \neg \exists q' < q. (x, q', i, *, \pi) \in \mathcal{O}_i^T.$$

(cf. Definition 8, which defines the maximum quantity, \bar{q} , as the largest of these.) For the divisible case, the distinguished quantities are those where the *unit price* changes, which can be formalized similarly.

To handle nonlinear pricing, we augment the network to include flow possibilities corresponding to each of the enumerated quantities, plus additional structure to enforce exclusivity among them. In other words, the network treats the offer for a given quantity as in Section 5.2, and embeds this in an XOR relation to ensure that each trader picks only one of these quantities. Since for each such quantity choice we can apply Theorem 3 or 4, the solution we get is in fact the solution to *GMAP*.

The network representation of the XOR relation (which can be embedded into the network of Figure 2) is depicted in Figure 3. For a trader i with K XOR quantity points, we define dummy variables, z_i^k , $k = 1, \dots, K$. Since we consider trades between every pair of quantity points we also have $q_{i_j}^k$, $k = 1, \dots, K$. For buyer $i \in AI$ with XOR points at quantities \bar{q}_i^k , we replace (4b) with the following constraints:

$$\begin{aligned} \sum_{j \in T(i)} q_{i_j}^k &= \bar{q}_i^k z_i^k \quad k = 1, \dots, K \\ \sum_{k=1}^K z_i^k &= r_i \\ z_i^k &\in \{0, 1\} \quad k = 1, \dots, K \end{aligned} \tag{5}$$

Figure 3 depicts the generalized network flow formulation.

5.4 Homogeneity Constraints

The model (4) handles constraints over the aggregation of quantities from different trading partners. When aggregation is allowed, the formulation permits trades involving arbitrary combinations of configurations. A *homogeneity constraint* [4] restricts such combinations, by requiring that configurations aggregated in an overall deal must agree on some or all attributes.

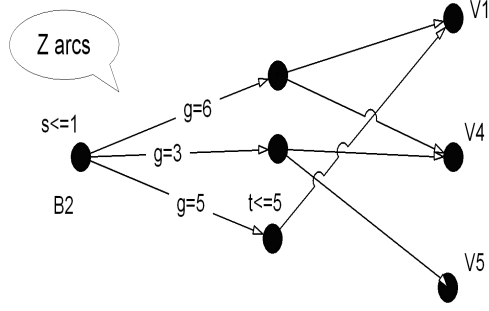


Figure 3: Extending the network flow model to express an XOR over quantities. B2 has 3 XOR points for 6, 3, or 5 units.

In the presence of homogeneity constraints, we can no longer apply the convenient separation of *GMAP* into *MMP* plus global bipartite optimization, as the solution to *GMAP* may include trades not part of any *MMP* solution. For example, let buyer b specify an offer for maximum quantity 10 of various acceptable configurations, with a homogeneity constraint over the attribute “color”. This means b is willing to aggregate deals over different trading partners and configurations, as long as all are the same color. If seller s can provide 5 blue units or 5 green units, and seller s' can provide only 5 green units, we may prefer that b and s trade on green units, even if the local surplus of a blue trade is greater.

Let $\{x_1, \dots, x_H\}$ be attributes that some trader constrains to be homogeneous. To preserve the network flow framework, we need to consider, for each trader, every point in the product domain of these homogeneous attributes. Thus, for every assignment \hat{x} to the homogeneous attributes, we compute *MMP*(b, s) under the constraint that configurations are consistent with \hat{x} . We apply the same approach as in Section 5.3: solve the global optimization, such that the alternative \hat{x} assignments for each trader are combined under XOR semantics, thus enforcing homogeneity constraints.

The size of this network is exponential in the number of homogeneous attributes, since we need a node for each point in the product domain of all the homogeneous attributes of each trader.⁶ Hence this solution method will only be tractable in applications where the traders can be limited to a small number of homogeneous attributes. It is important to note that the graph needs to include a node only for each point that potentially matches a point of the other side. It is therefore possible to make the problem tractable by limiting one of the sides to a less expressive bidding language, and by that limit the set of potential matches. For example, if sellers submit bounded sets of XOR points, we only need to consider the points in the combined set offered by the sellers, and the reduction to network flow is polynomial regardless of the number of homogeneous attributes.

If such simplifications do not apply, it may be preferable to solve the global problem directly as a single optimization problem. We provide the formulation for the special case of divisibility (with respect to integers) and configuration parity. Let i index buyers, j sellers, and H homogeneous attributes. Variable $x_{i,j}^h \in X_h$ represents the value of attribute X_h in the trade between buyer i and seller j . Integer variable $q_{i,j}$ represents the quantity of the trade (zero for no trade) between i and j .

⁶If traders differ on which attributes they express such constraints, we can limit consideration to the relevant alternatives. The complexity will still be exponential, but in the maximum number of homogeneous attributes for any pair of traders.

$$\begin{aligned}
 \max \quad & \sum_{i \in B, j \in S} [\hat{u}_i^B(x_{i,j}, q_{i,j}) - \hat{u}_j^S(x_{i,j}, q_{i,j})] \\
 \sum_{j \in S} \quad & q_{i,j} \leq \bar{q}_i \quad i \in B \\
 \sum_{i \in B} \quad & q_{i,j} \leq \bar{q}_j \quad j \in S \\
 x_{1_j}^h = x_{2_j}^h = \dots = x_{|B|_j}^h \quad & j \in S, h \in \{1, \dots, H\} \\
 x_{i_1}^h = x_{i_2}^h = \dots = x_{i_{|S|}}^h \quad & i \in B, h \in \{1, \dots, H\}
 \end{aligned} \tag{6}$$

Table 1 summarizes the mapping we presented from allocation constraints to the complexity of solving *GMAP*. Configuration parity is assumed for all cases but the first.

6. EXPERIMENTAL RESULTS

We approach the experimental aspect of this work with two objectives. First, we seek a general idea of the sizes and types of clearing problems that can be solved under given time constraints. We also look to compare the performance of a straightforward integer program as in (4) with an integer program that is based on the network formulations developed here. Since we used CPLEX, a commercial optimization tool, the second objective could be achieved to the extent that CPLEX can take advantage of network structure present in a model.

We found that in addition to the problem size (in terms of number of traders), the number of aggregating traders plays a crucial role in determining complexity. When most of the traders are aggregating, problems of larger sizes can be solved quickly. For example, our IP model solved instances with 600 buyers and 500 sellers, where 90% of them are aggregating, in less than two minutes. When the aggregating ratio was reduced to 80% for the same data, solution time was just under five minutes.

These results motivated us to develop a new network model. Rather than treat non-aggregating traders as a special case, the new model takes advantage of the single-unit nature of non-aggregating trades (treating the aggregating traders as a special case). This new model outperformed our other models on most problem instances, exceptions being those where aggregating traders constitute a vast majority (at least 80%).

This new model (Figure 4) has a single node for each non aggregating trader, with a single-unit arc designating a match to another non-aggregating trader. An aggregating trader has a node for each potential match, connected (via y arcs) to a mutual source node. Unlike the previous model we allow fractional flow for this case, representing the traded fraction of the buyer’s total quantity.⁷

We tested all three models on random data in the form of bipartite graphs encoding *MMP* solutions. In our experiments, each trader has a maximum quantity uniformly distributed over $[30, 70]$, and minimum quantity uniformly distributed from zero to maximal quantity. Each buyer/seller pair is selected as matching with probability 0.75, with matches assigned a surplus uniformly distributed over $[10, 70]$. Whereas the size of the problem is defined by the number of traders on each side, the problem complexity depends on the product $|B| \times |S|$. The tests depicted in Figures 5–7 are for the worst case $|B| = |S|$, with each data point averaged over six samples. In the figures, the direct IP (4) is designated “SW”, our first network model (Figure 2) “NW”, and our revised network model (Figure 4) “NW 2”.

⁷Traded quantity remains integer.

Aggregation	Hom. attr.	Divisibility	linear pricing	Technique	Complexity
No aggregate	N/A	Any	Not required	Assignment problem	Polynomial
All aggregate	None	Down to 0	Required	Transpor. problem	Polynomial
One side	None	Aggr side div.	Aggr. side	GAP	NP-hard
Optional	None	Down to q , batch	Required	Generalized ntwrk flow	NP-hard
Optional	Bounded	Down to q , batch	Bounded size schdl.	Generalized ntwrk flow	NP-hard
Optional	Not bounded	Down to q , batch	Not required	Nonlinear opt	Depends on $\hat{u}(x, q)$

Table 1: Mapping from combinations of allocation constraints to the solution methods of GMAP. One Side means that one side aggregates and divisible, and the other side is restrictive. *Batch* means that traders may submit batch sizes.

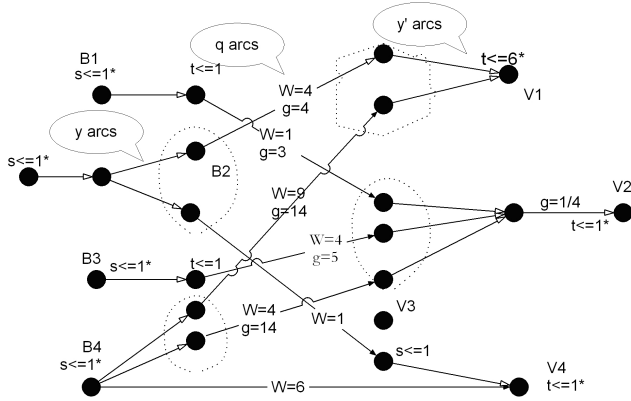


Figure 4: Generalized network flow model. $B1$ is a buyer in AD , $B2 \in AI$, $B3 \in NI$, $B4 \in ND$. $V1$ is a seller in AD , $V2 \in AI$, $V4 \in ND$. The g values represent arc gains, and W values represent weights.

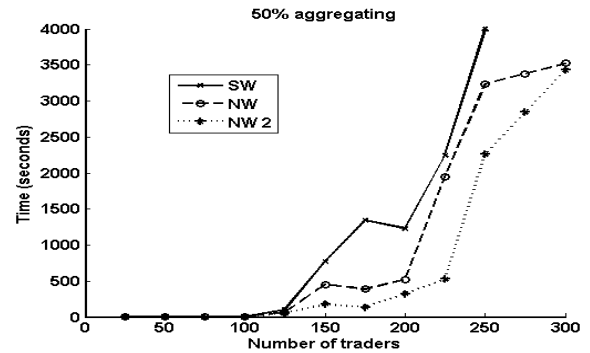


Figure 6: Average performance of models when 50% of traders aggregate.

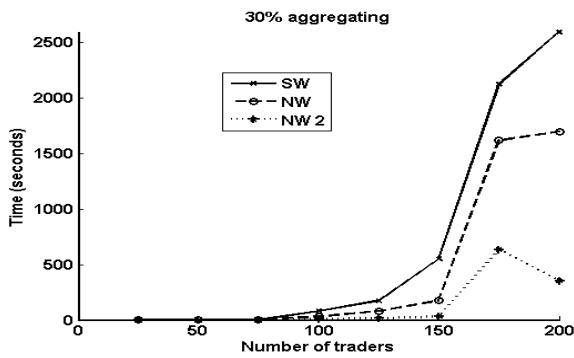


Figure 5: Average performance of models when 30% of traders aggregate.

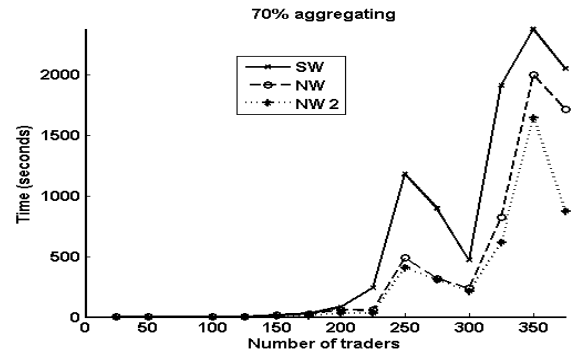


Figure 7: Average performance of models when 70% of traders aggregate.

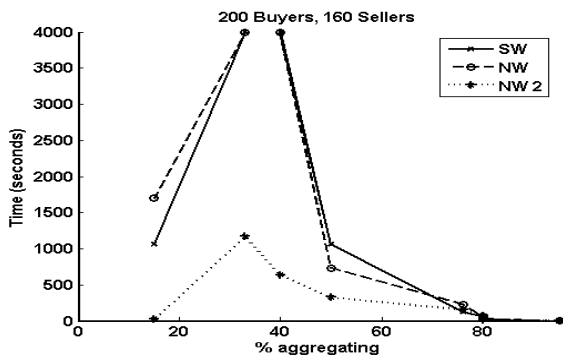


Figure 8: Performance of models when varying percentage of aggregating traders

Figure 8 shows how the various models are affected by a change in the percentage of aggregating traders, holding problem size fixed.⁸

Due to the integrality constraints, we could not test available algorithms specialized for network-flow problems on our test problems. Thus, we cannot fully evaluate the potential gain attributable to network structure. However, the model we built based on the insight from the network structure clearly provided a significant speedup, even without using a special-purpose algorithm. Model NW 2 provided speedups of a factor of 4–10 over the model SW. This was consistent throughout the problem sizes, including the smaller sizes for which the speedup is not visually apparent on the chart.

7. CONCLUSIONS

The implementation and deployment of market exchanges requires the development of bidding languages, information feedback policies, and clearing algorithms that are suitable for the target domain, while paying heed to the incentive properties of the resulting mechanisms. For multiattribute exchanges, the space of feasible such mechanisms is constrained by computational limitations imposed by the clearing process. The extent to which the space of feasible mechanisms may be quantified a priori will facilitate the search for such exchanges in the full mechanism design problem.

In this work, we investigate the space of two-sided multiattribute auctions, focusing on the relationship between constraints on the offers traders can express through bids, and the resulting computational problem of determining an optimal set of trades. We developed a formal semantic framework for characterizing expressible offers, and introduced some basic classes of restrictions. Our key technical results identify sets of conditions under which the overall matching problem can be separated into first identifying optimal pairwise trades and subsequently optimizing combinations of those trades. Based on these results, we developed network flow models for the overall clearing problem, which facilitate classification of problem versions by computational complexity, and provide guidance for developing solution algorithms and relaxing bidding constraints.

8. ACKNOWLEDGMENTS

This work was supported in part by NSF grant IIS-0205435, and the STIET program under NSF IGERT grant 0114368. We are

⁸All tests were performed on Intel 3.4 GHz processors with 2048 KB cache. Test that did not complete by the one-hour time limit were recorded as 4000 seconds.

grateful to comments from an anonymous reviewer. Some of the underlying ideas were developed while the first two authors worked at TradingDynamics Inc. and Ariba Inc. in 1999–2001 (cf. US Patent 6,952,682). We thank Yoav Shoham, Kumar Ramaiyer, and Gopal Sundaram for fruitful discussions about multiattribute auctions in that time frame.

9. REFERENCES

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice-Hall, 1993.
- [2] F. Bacchus and A. Grove. Graphical models for preference and utility. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 3–10, Montreal, 1995.
- [3] M. Bichler. *The Future of e-Markets: Multi-Dimensional Market Mechanisms*. Cambridge U. Press, New York, NY, USA, 2001.
- [4] M. Bichler and J. Kalagnanam. Configurable offers and winner determination in multi-attribute auctions. *European Journal of Operational Research*, 160:380–394, 2005.
- [5] M. Bichler, M. Kaukal, and A. Segev. Multi-attribute auctions for electronic procurement. In *Proceedings of the 1st IBM IAC Workshop on Internet Based Negotiation Technologies*, 1999.
- [6] C. Boutilier, T. Sandholm, and R. Shields. Eliciting bid taker non-price preferences in (combinatorial) auctions. In *Nineteenth Natl. Conf. on Artificial Intelligence*, pages 204–211, San Jose, 2004.
- [7] F. Branco. The design of multidimensional auctions. *RAND Journal of Economics*, 28(1):63–81, 1997.
- [8] Y.-K. Che. Design competition through multidimensional auctions. *RAND Journal of Economics*, 24(4):668–680, 1993.
- [9] G. Debreu. Topological methods in cardinal utility theory. In K. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical Methods in the Social Sciences*. Stanford University Press, 1959.
- [10] N. Economides and R. A. Schwartz. Electronic call market trading. *Journal of Portfolio Management*, 21(3), 1995.
- [11] E. Fink, J. Johnson, and J. Hu. Exchange market for complex goods: Theory and experiments. *Netnomics*, 6(1):21–42, 2004.
- [12] M. L. Fisher, R. Jaikumar, and L. N. Van Wassenhove. A multiplier adjustment method for the generalized assignment problem. *Management Science*, 32(9):1095–1103, 1986.
- [13] J. Gong. Exchanges for complex commodities: Search for optimal matches. Master’s thesis, University of South Florida, 2002.
- [14] C. Gonzales and P. Perny. GAI networks for decision making under certainty. In *IJCAI-05 Workshop on Advances in Preference Handling*, Edinburgh, 2005.
- [15] J. R. Kalagnanam, A. J. Davenport, and H. S. Lee. Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand. *Electronic Commerce Research*, 1(3):221–238, 2001.
- [16] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, 1976.
- [17] N. Nisan. Bidding and allocation in combinatorial auctions. In *Second ACM Conference on Electronic Commerce*, pages 1–12, Minneapolis, MN, 2000.
- [18] D. C. Parkes and J. Kalagnanam. Models for iterative multiattribute procurement auctions. *Management Science*, 51:435–451, 2005.
- [19] T. Sandholm and S. Suri. Side constraints and non-price attributes in markets. In *IJCAI-01 Workshop on Distributed Constraint Reasoning*, Seattle, 2001.
- [20] L. J. Schwartzman and M. P. Wellman. Market-based allocation with indivisible bids. In *AAMAS-05 Workshop on Agent-Mediated Electronic Commerce*, Utrecht, 2005.
- [21] J. Shachat and J. T. Swarhout. Procurement auctions for differentiated goods. Technical Report 0310004, Economics Working Paper Archive at WUSTL, Oct. 2003.
- [22] A. V. Sunderam and D. C. Parkes. Preference elicitation in proxied multiattribute auctions. In *Fourth ACM Conference on Electronic Commerce*, pages 214–215, San Diego, 2003.
- [23] P. R. Wurman, M. P. Wellman, and W. E. Walsh. A parametrization of the auction design space. *Games and Economic Behavior*, 35:304–338, 2001.