Side Constraints and Non-Price Attributes in Markets[*]

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
sandholm@cs.cmu.edu

Subhash Suri
Department of Computer Science
University of California
Santa Barbara, CA 93106
suri@cs.ucsb.edu

Abstract

In most real-world (electronic) marketplaces, there are other considerations besides maximizing immediate economic value. We present a sound way of taking such considerations into account via side constraints and non-price attributes. Side constraints have a significant impact on the complexity of market clearing. Budget constraints, a limit on the number of winners, and XOR-constraints make even noncombinatorial markets $\mathcal{NP}$-complete to clear. The latter two make markets $\mathcal{NP}$-complete to clear even if bids can be accepted partially. This is surprising since, as we show, even combinatorial markets with a host of very similar side constraints can be cleared in polytime. An extreme equality constraint makes combinatorial markets polytime clearable even if bids have to be accepted entirely or not at all. Finally, we present a way to take into account additional attributes using a bid re-weighting scheme, and prove that it does not change the complexity of clearing. All of the results hold for auctions as well as exchanges, with and without free disposal.

1 Introduction

For a long time in the AI community, auctions and exchanges have been proposed as mechanisms for allocating items (resources, tasks, etc.) in multiagent systems consisting of self-interested parties. Some of the market mechanisms that lead to economically efficient outcomes among the parties are computationally complex to clear. In particular, there has been a recent surge of interest in algorithms for clearing auctions where bids can be submitted on combinations of items [8; 9; 1; 4; 10; 11; 3; 15; 5; 12; 13]. To our knowledge, all of that literature has focused on clearing combinatorial auctions so as to maximize unconstrained economic value. In most real-world marketplaces, especially in business-to-business commerce, there are other considerations besides maximizing immediate economic value that must be taken into account.

In this paper we introduce and analyze two methods for incorporating these additional considerations: side constraints on the trading outcome, and bid re-weighting with multivariate functions to include non-price attributes. In short, our goal is to develop market designs that are computationally tractable to clear, and where true economic value is maximized, taking into account all the pertinent constraints and attributes of the market. The side constraints could be imposed by any party: the buyer(s), the seller(s), the party who executes the marketplace, the party who develops the technology for clearing the market, a regulatory body such as the SEC, liquidity providers (such as *market makers* on the NASDAQ or *specialists* on the NYSE), etc. Similarly, the set of pertinent attributes and their values could be specified by any party.

The prior literature has focused on the settings where bids must be accepted as a whole or rejected, while we also cover the important practical setting where bids can be partially accepted. The literature has also focused on settings with *free disposal* (sellers can keep any item and/or buyers can take extra units of any item). We analyze markets both with and without free disposal.

We first discuss side constraints in markets where bids are on individual items, and then move to markets where bids can be submitted on combinations of items. Finally, we show how to integrate non-price attributes into (combinatorial) market designs.

2 Singleton Bids

In this section we show that certain practical side constraints can make even noncombinatorial auctions hard to clear.

Definition 1 (WDP) *The seller has m items (one unit each) to sell. Each bidder places a set of bids on individual items. The* winner determination problem (WDP) *is to determine a revenue-maximizing allocation of items to bidders.*

In the absence of side constraints, WDP can be solved in polytime by picking the highest bid for each item independently. The budget constraint below illustrates how sharp the $\mathcal{P}$ vs. $\mathcal{NP}$-complete cutoff is in the space of side constraints. This is especially surprising since a similar constraint, where the *number of items sold to each bidder* is constrained, leads to a WDP that was recently shown to be polytime solvable using *b-matching* [15].[1]

[*]This work was funded by, and conducted at, CombineNet, Inc., 311 S. Craig St., Pittsburgh, PA 15213.

[1]Multi-item auctions (with bids on individual items only) with certain types of structural side constraints are also solvable in polytime using b-matching [6].

Definition 2 (BUDGET) *WDP where the amount sold to any bidder does not exceed her budget.*[2]

Theorem 2.1 *BUDGET is $\mathcal{NP}$-complete (whether or not the seller has to sell all the items), even with integer prices.*

PROOF. We reduce PARTITION [2] to BUDGET. In PARTITION, we have a set of integers $S = \{x_1, x_2, \ldots, x_n\}$, and the goal is to partition S into two subsets A and B (i.e. $A \cap B = \emptyset$ and $A \cup B = S$) s.t.

$$z = \sum_{i \in A} x_i = \sum_{i \in B} x_i,$$

where $z = \frac{1}{2} \sum_{x \in S} x$. We create an instance of BUDGET as follows. Corresponding to each x_i, we create an item i. There are two bidders, say, Andy and Bob; each places the bid of same price, x_i, for item i. The budget for Andy and Bob each is z (half of the total). This instance of BUDGET has a solution with revenue $2z$ if and only if the original partition problem has a partition. □

If bids can be accepted partially, BUDGET can be solved in polytime using linear programming.

Another practical side constraint is the number of winners. For example, the seller may not want the overhead of dealing with a large number of winning bidders.

Definition 3 (MAX-WINNERS) *WDP where at most k bidders receive items.*

Theorem 2.2 *MAX-WINNERS is $\mathcal{NP}$-complete (whether or not the seller has to sell all the items), even with integer prices.*

PROOF. We reduce SET-COVER [2] to MAX-WINNERS. Given an instance of set cover, namely, a ground set $X = \{1, 2, \ldots, m\}$, and a set of subsets $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$, where $S_i \subseteq X$, we formulate an instance of MAX-WINNERS as follows. We create an item i for each element i in the ground set X. Corresponding to each set S_i, we create a bidder B_i, who places a \$1 bid on each item in the set S_i.

We claim that there is a set cover of size k if and only if the auction has a solution with revenue m and max number of winners k.

- [$\Rightarrow$] Consider a feasible solution for the auction. We claim that the sets corresponding to the k winning bidders form a set cover. (That is, if bidder i receives at least one item, then we put the set S_i in the cover.) Since the revenue is m, each item must be awarded to some bidder, and hence it must be covered by the set cover.
- [$\Leftarrow$] Consider a solution to the set cover. For each set S_i in the cover, we make bidder i a winner. Since each item is covered in the set cover, each item is bid upon by at least one bidder in the just constructed winning set, but the item may be claimed by more than one winning bidders. However, since each bid is for the same price, we can arbitrarily award the item to any of the winning bidders claiming this item. This gives a solution to the auction problem with revenue m and number of winning bidders k. □

[2]Budget constraints occur naturally in markets, and they have been studied from the *bidding* perspective in the literature before [7].

In many problems, allowing the decision variables to be continuous instead of binary causes the problem to become polytime solvable. For example, the KNAPSACK problem [2] becomes trivial to solve optimally if packages can be included partially (simply accept packages in descending value-to-weight order, the last one partially). However, the MAX-WINNERS problem remains hard:

Theorem 2.3 *Even if bids can be accepted partially, MAX-WINNERS is $\mathcal{NP}$-complete (whether or not the seller has to sell all the items), even with integer prices.*

PROOF. The reduction used to prove Theorem 2.2 applies. Whenever a bid is accepted even partially, the corresponding set is included in the cover. □

In some settings, a bidder may want to submit bids on multiple items, but may want to mutually exclude some of the items. For example, a buyer may want to buy a VCR and a TV, and either of two TVs (but not both) would be acceptable. She could express this by placing bids on each of the three items, but inserting an XOR-constraint between the bids on the TVs.

Definition 4 (XORS) *WDP with XOR-constraints. Whenever two bids are combined with XOR, at most one of them can win.*

Theorem 2.4 *XORS is $\mathcal{NP}$-complete (whether or not the seller has to sell all the items), even with integer prices.*

PROOF. We reduce INDEPENDENT-SET [2] to XORS. Corresponding to each vertex, generate an item and a \$1 bid for that item. Corresponding to each edge, insert an XOR-constraint between the bids. Now, XORS has a solution of \$k iff there is an independent set of size k. □

Theorem 2.5 *Even if bids can be accepted partially, XORS is $\mathcal{NP}$-complete (whether or not the seller has to sell all the items), even with integer prices.*

PROOF. The proof of Theorem 2.4 applies. Whenever a bid is partially accepted in an auction, it might as well be completely accepted since its neighbors are not accepted anyway. □

3 Combinatorial Bidding and Asking

In this section we define combinatorial auctions and multi-unit combinatorial exchanges. In the next section we show how side constraints affect the complexity of clearing these markets.

3.1 Combinatorial Auctions

In a *combinatorial auction (CA)*, bidders may submit bids on combinations of items. This allows the bidders to express the fact that the value of a bundle of items may differ from the sum of the values of the individual items that constitute the bundle.

Definition 5 (CAWDP) *The auctioneer has a set of items, $M = \{1, 2, \ldots, m\}$, to sell, and the buyers submit a set of bids, $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$. A bid is a tuple $B_j = \langle S_j, p_j \rangle$, where $S_j \subseteq M$ is a set of items and p_j, $p_j \geq 0$ is a price. The* combinatorial auction winner determination problem (CAWDP) *is to label the bids as winning or losing so* *as to maximize the auctioneer's revenue under the constraint that each item can be allocated to at most one bidder:*

$$\max \sum_{j=1}^{n} p_j x_j \quad \textit{s.t.} \quad \sum_{j|i\in S_j} x_j \leq 1 \qquad i = 1, 2, \ldots, m$$

If there is no free disposal (auctioneer is not willing to keep any of the items, and bidders are not willing to take extra items), an equality is used in place of the inequality.

Definition 6 (BCAWDP) *Binary CAWDP: CAWDP where the decision variables are binary ($x_j \in \{0,1\}$, i.e., each bid has to be completely accepted or not at all).*

Even with free disposal and integer prices, BCAWDP is $\mathcal{NP}$-complete [8], and it cannot even be approximated to a ratio of $n^{1-\epsilon}$ in polytime (unless $\mathcal{P} = \mathcal{NP}$) [9].

Definition 7 (CCAWDP) *Continuous CAWDP: CAWDP where the decision variables are continuous ($0 \leq x_j \leq 1$, i.e., bids can be accepted partially).*

CCAWDP is directly solvable by linear programming in polytime. While academic research on clearing combinatorial auctions has focused on the binary case [8; 9; 1; 10; 11; 15; 3; 5], the currently biggest real combinatorial markets are continuous. For example, when `logistics.com` auctions long-term trucking lanes (the volume of each lane is numerous truck-loads), carriers can bid on combinations of lanes, and bids can be accepted partially. Also, in the `BondConnect` combinatorial bond exchange, bids can be accepted partially.

3.2 Combinatorial Reverse Auctions

Next we introduce a combinatorial reverse auction.

Definition 8 (CRAWDP) *The buyer wants to obtain a set of items, $M = \{1, 2, \ldots, m\}$, and the sellers submit a set of asks, $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$. An ask is a tuple $B_j = \langle S_j, p_j\rangle$, where $S_j \subseteq M$ is a set of items and p_j, $p_j \geq 0$ is an asking price. The* combinatorial reverse auction winner determination problem (CRAWDP) *is to label the asks as winning or losing so as to minimize the buyer's cost under the constraint that the buyer obtains each item:*

$$\min \sum_{j=1}^{n} p_j x_j \quad \textit{s.t.} \quad \sum_{j|i\in S_j} x_j \geq 1 \qquad i = 1, 2, \ldots, m$$

If there is no free disposal (the buyer cannot take extra units and sellers cannot keep any of the items of their winning asks), an equality is used in place of the inequality.

Definition 9 (BCRAWDP) *Binary CRAWDP: CRAWDP where the decision variables are binary ($x_j \in \{0,1\}$, i.e., each ask has to be completely accepted or not at all).*

Proposition 3.1 *BCRAWDP is $\mathcal{NP}$-complete, even with free disposal.*

PROOF. BCRAWDP with free disposal is equivalent to weighted set covering, which is $\mathcal{NP}$-complete. □

Definition 10 (CCRAWDP) *Continuous CRAWDP: CRAWDP where the decision variables are continuous ($0 \leq x_j \leq 1$, i.e., asks can be accepted partially).*

CCRAWDP is directly solvable by linear programming in polytime.

3.3 Multi-Unit Combinatorial Exchanges

In a multi-unit combinatorial exchange, both buyers and sellers can submit combinatorial bids, and in one bid, a bidder might be selling units of some items and buying units of other items [10; 11].

Definition 11 (MUCEWDP) *A bid in this setting is $B_j = \langle(\lambda_j^1, \lambda_j^2, \ldots \lambda_j^m), p_j\rangle$, where $\lambda_j^k \in \Re$ is the requested number of units of item k, and $p_j \in \Re$ is the price. A positive λ_j^k represents buying and a negative λ_j^k means selling. A positive p_j represents bidding while a negative p_j means asking. The* multi-unit combinatorial exchange winner determination problem (MUCEWDP) *is to label the bids as winning or losing so as to maximize surplus under the constraint that demand does not exceed supply:*

$$\max \sum_{j=1}^{n} p_j x_j \quad \textit{s.t.} \quad \sum_{j=1}^{n} \lambda_j^i x_j \leq 0 \qquad i = 1, 2, \ldots, m$$

If there is no free disposal (buyers are not willing to take extra units, and sellers are not willing to keep any units of their winning bids), an equality is used in place of the inequality.

Definition 12 (BMUCEWDP) *Binary MUCEWDP: MUCEWDP where the decision variables are binary ($x_j \in \{0,1\}$, i.e., each bid has to be completely accepted or not at all).*

Proposition 3.2 *BMUCEWDP is $\mathcal{NP}$-complete (with and without free disposal), even with integer prices and units. It is also inapproximable in polynomial time (unless $\mathcal{P} = \mathcal{NP}$).*

PROOF. BCAWDP is a special case of BMUCEWDP. □

Definition 13 (CMUCEWDP) *Continuous MUCEWDP: MUCEWDP where the decision variables are continuous ($0 \leq x_j \leq 1$, i.e., bids and asks can be accepted partially).*

CMUCEWDP is directly solvable by linear programming in polytime.

3.4 Spanning the Spectrum of Combinatorial Market Designs

In the next section we analyze the complexity of winner determination under different side constraints. We present the positive results in the context of the most general combinatorial market design (MUCEWDP). They therefore apply to all special cases of it as well, such as CAWDP and CRAWDP. We present the negative results in the context of CAWDP and CRAWDP. They therefore apply to all generalizations thereof as well, such as MUCEWDP.

4 Side Constraints in Combinatorial Markets

In this section we discuss how the complexity of clearing a combinatorial market changes as different types of side constraints are imposed on the outcome. It turns out that different side constraints introduce sharp cutoffs in the complexity of clearing. Seemingly similar side constraints lead to problems that lie on different sides of the $\mathcal{P}$ vs. $\mathcal{NP}$-complete cutoff. In the first subsection we present side constraints under which the continuous case remains easy and the binary case remains hard. In the next subsection we present side constraints that make both cases hard. In the last subsection we present a side constraint that make both cases easy.

4.1 Side Constraints under which the Continuous Case Remains Easy, and the Binary Case Remains Hard

The following classes of domain-independent side constraints, which we view as practically important and quite general, turn out to be easy for the continuous winner determination problem, and remain hard for the binary case. The constraint classes may seem cumbersome. That is because we focused on making them as general as possible. Each constraint class encompasses several types of simpler practical constraints, as we will discuss via examples.

The constraints use the following terminology. Let the *net revenue (NR)* of a set of bids X be $-\sum_{j\in X} p_j x_j$ (bids decrease NR, but asks increase NR because the prices are negative). Let the *gross revenue (GR)* of a set of bids X be $\sum_{j\in X} |p_j| x_j$ (both bids and asks increase GR). Let the *net units (NU)* of a set of bids X and set of items Y be $\sum_{j\in X}\sum_{i\in Y} \lambda_j^i x_j$ (units bought increase NU but units sold are negative, so they decrease NU). Let the *gross units (GU)* of a set of bids X and set of items Y be $\sum_{j\in X}\sum_{i\in Y} |\lambda_j^i| x_j$ (units bought and sold increase GU). Roughly, the gross terms measure market share and the net terms measure property obtained. The revenue terms measure these in money received, while the unit terms measure these in goods received. On all four terms, each player prefers a high value.

Maximum trade constraints:

1. MAX-SUBSET-NET-REVENUE: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), NR $\le \$k$. For example, this can be used to enforce that a seller does not get a net revenue that is obscenely high (which could be considered out of line).
2. %MAX-SUBSET-NET-REVENUE: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), NR cannot exceed k% of the NR from set $\mathcal{B}'' \subseteq \mathcal{B}$. A current large-scale real-world market that runs combinatorial auctions has to guarantee that 30% of the dollar value of the awarded bids goes to minority bidders. This could be implemented using the %MAX-SUBSET-NET-REVENUE constraint (because revenues of buyers are nonpositive). As another example, to maintain fairness, a buyer in a reverse auction may not want any seller to get more than a certain fraction of the revenue that the market generates.
3. MAX-SUBSET-GROSS-REVENUE: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), GR $\le \$k$. This can be used, for example, to guarantee that a certain class of buyers gets a certain dollar volume of a market.
4. %MAX-SUBSET-GROSS-REVENUE: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), GR cannot exceed k% of the GR from set $\mathcal{B}'' \subseteq \mathcal{B}$. This can be used, for example, to guarantee a certain class of buyers a given percentage of the market share. That allows the marketplace to enforce *diversity* on the buyer side, which may hedge against risks such as failure to pay.
5. MAX-SUBSET-NET-UNITS: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), and a certain set of items $M' \subseteq M$, NU $\le k$. This can be used, for example, to ensure that no buyer gets an obscenely large number of units, or to guarantee that a seller gets to sell at least a certain number of units of some items.
6. %MAX-SUBSET-NET-UNITS: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), and a certain set of items $M' \subseteq M$, NU cannot exceed k% the NU of items $M'' \subseteq M$ from bid set $\mathcal{B}'' \subseteq \mathcal{B}$. This can be used, for example, to ensure that no buyer gets an obscenely large fraction of units, or to guarantee that a seller gets to sell at least a certain fraction of units.
7. MAX-SUBSET-GROSS-UNITS: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), and a certain set of items $M' \subseteq M$, GU $\le k$. For example, this can be used in an exchange to guarantee a maximum trading volume to a buyer, or a minimum for a seller.
8. %MAX-SUBSET-GROSS-UNITS: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain player or by a set of players), and a certain set of items $M' \subseteq M$, GU cannot exceed k% of the GU of items $M'' \subseteq M$ from bid set $\mathcal{B}'' \subseteq \mathcal{B}$. For example, this can be used in an exchange to guarantee a maximum fraction of trading volume to a buyer, or a minimum for a seller.

Minimum trade constraints:

1. An analogous MIN-constraint to each of the eight MAX-constraints above can be derived directly by turning the inequality around. For example, a buyer with a budget constraint is a special case of MIN-SUBSET-NET-REVENUE. For a more general example of MIN-SUBSET-NET-REVENUE, consider a firm that bids on behalf of its different business units, and each business unit has its own purchasing budget that cannot be exceeded. This would induce one constraint per business unit. The MIN-SUBSET-NET-REVENUE constraint can also be used to guarantee that a seller gets at least a certain net revenue.

   As another example, the %MIN-SUBSET-GROSS-REVENUE constraint could be used to enforce that a certain set of sellers gets at least a given fraction of the market share. Placing such constraints allows the marketplace to enforce *diversity* on the seller side, which may hedge against risks such as nondelivery. It could also be used to guarantee that one player does not get more than twice the volume of another player.

Minimum mutual trade constraints:

1. MUTUAL-TRADE: Set A of sellers of item i must sell to set B of buyers of i at least k units of i. One of the main arguments against dynamic pricing has been the desirability of stable long-term business relationships with trade volume guarantees. The MUTUAL-TRADE constraint allows that concern to be incorporated into a dynamically-priced marketplace. This enables the participants to capture the advantages of both long-term business guarantees and dynamic pricing.

2. %MUTUAL-TRADE: Set A of sellers of item i must sell to set B of buyers of i at least k% of the units of item i sold by set C of sellers (C may be a subset of A, a superset of A, intersect with A, or be disjoint from A). The motivation is the same as for MUTUAL-TRADE. (If C is the set of all sellers, then this constraint forces a certain fraction of the trade on item i to be conducted directly between A and B.) [3]

Minimum constraints on trading on all items:

1. TRADE-ON-ALL-ITEMS: The items in set $M' \subseteq M$ have to trade a total of at least k units.
2. %TRADE-ON-ALL-ITEMS: The items in set $M' \subseteq M$ have to trade a total of at least k% of the units traded of some other set of items $M'' \subseteq M$.

MAX/MIN constraints on acceptance ratio:

1. MAX-SUBSET-ACCEPTANCE-RATIO: Of a certain set $\mathcal{B}' \subseteq \mathcal{B}$ of bids and asks (for example, by a certain bidder or by a set of bidders), at most a certain ratio can be accepted: $\frac{\sum_{j|B_j \in \mathcal{B}'} x_j}{|\mathcal{B}'|} \leq k\%$.
2. MIN-SUBSET-ACCEPTANCE-RATIO: Of a certain set $\mathcal{B}' \subseteq \mathcal{B}$ of bids and asks (for example, by a certain bidder or by a set of bidders), at least a certain ratio has to be accepted. This could be used to mitigate the frustration of losing, and to induce more bidding.

The remaining constraints are equalities. They are designed to enforce strong forms of fairness among the market participants in terms of equitable allocation. The equalities can be taken among buyers, among sellers, or across buyers and sellers (and some players may both buy and sell—even in the same bid). As is, comparing the net measures across buyers and sellers only makes sense in a barter economy since the net measures for those two sets generally have different signs. This can be generalized directly by comparing absolute values of the net measures. The theorems in this section apply to that case as well.

General equality constraints on trading volume:

1. EQUAL-SUBSET-NET-REVENUE: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain bidder or by a set of bidders), NR equals the NR of another bid set $\mathcal{B}'' \subseteq \mathcal{B}$.
2. EQUAL-SUBSET-GROSS-REVENUE: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain bidder or by a set of bidders), GR equals the GR of another bid set $\mathcal{B}'' \subseteq \mathcal{B}$.
3. EQUAL-SUBSET-NET-UNITS: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain bidder or by a set of bidders), and a certain set of items $M' \subseteq M$, NU equals the NU of another bid set $\mathcal{B}'' \subseteq \mathcal{B}$.
4. EQUAL-SUBSET-GROSS-UNITS: Of a certain set of bids $\mathcal{B}' \subseteq \mathcal{B}$ (for example, by a certain bidder or by a set of bidders), and a certain set of items $M' \subseteq M$, GU equals the GU of another bid set $\mathcal{B}'' \subseteq \mathcal{B}$.

Strict equality constraints on trading volume:

1. EQUAL-NET-REVENUE: Each bidder gets equal NR.
2. EQUAL-GROSS-REVENUE: Each bidder gets equal GR.
3. EQUAL-NET-UNITS: Each bidder gets equal NU on a given set of items $M' \subseteq M$.
4. EQUAL-GROSS-UNITS: Each bidder gets equal GU on a given set of items $M' \subseteq M$.

Strict equality constraints on acceptance ratio:

1. EQUAL-NET-REVENUE-ACCEPTANCE-RATIO: Every bidder gets awarded the same ratio of the net revenue of her bids (NR / NR as if all her bids got accepted).[4]
2. EQUAL-GROSS-REVENUE-ACCEPTANCE-RATIO: Every bidder gets awarded the same ratio of the gross revenue of her bids (GR / GR as if all her bids got accepted).
3. EQUAL-NET-UNITS-ACCEPTANCE-RATIO: Every bidder gets awarded the same ratio of the net units of her bids (NU / NU as if all her bids got accepted) on a given set of items $M' \subseteq M$.
4. EQUAL-GROSS-UNITS-ACCEPTANCE-RATIO: Every bidder gets awarded the same ratio of the gross units of her bids (GU / GU as if all her bids got accepted) on a given set of items $M' \subseteq M$.

In the constraints above, when NR, GR, NU, and GU comparisons are made, they are made within the same type (e.g., NU against NU). The following theorems would apply to constraints where comparisons are made across these types as well (although we believe that such constraints are less likely to be of relevance in practice). Furthermore, they would apply to constraints on the difference between the gross and net measures, as well as to constraints on the ratio of the gross and net measures.

Theorem 4.1 *BCAWDP (and CRAWDP) with constraints from any of the classes presented in this section is $\mathcal{NP}$-complete (with and without free disposal), even with integer prices.*

PROOF. By appropriately picking the parameters for the MAX and MIN constraints, they can be relaxed so they do not constrain the set of feasible allocations. BCAWDP with such constraints is therefore rich enough to emulate BCAWDP itself, which is $\mathcal{NP}$-complete.

To see that BCAWDP with the equality constraints is $\mathcal{NP}$-complete, consider a CA with just one bidder. The equalities do not bind, but the problem still is weighted set packing, which is $\mathcal{NP}$-complete. □

[3] Under the mutual business constraints, the resulting allocation ensures that it is possible to have the desired amount of trade between A and B. However, the basic market mechanism only says how much each party trades, not with whom. A postprocessor can be used to enforce that the minimum desired amount of trade is conducted between A and B directly.

[4] These four constraints can sometimes be too strong in the sense that a player's own bid can preclude the acceptance of some of her other bids because her bids share items.

Theorem 4.2 *CMUCEWDP with constraints from any of the classes presented in this section is polytime solvable (with and without free disposal).*

PROOF. Each constraint from any one of these classes can be modeled as one row (constraint) that is added to the linear program (we skip these encodings due to restricted space, but they are not extremely hard to construct). The resulting linear program is therefore of polynomial size in the size of the input. Linear programs can be solved in polynomial time in the size of the linear program (using interior point methods). □

4.2 Side Constraints under which the Continuous and Binary Case Are Hard

The most interesting results of this paper show that some classes of side constraints that are among the most important ones from a practical perspective, make even the continuous case $\mathcal{NP}$-complete to clear.

Theorem 4.3 *If no more than k winners are allowed, BCAWDP and CCAWDP (as well as BCRAWDP and CCRAWDP) are $\mathcal{NP}$-complete (with and without free disposal), even if prices are integer.*

PROOF. By Theorems 2.2 and 2.3, even the special case where the bids are all on singletons is $\mathcal{NP}$-complete. □

In a combinatorial auction where the bids are combined with OR, a bidder can only express complementarity, not substitutability. For example, say a bidder has submitted three bids: $\langle\{1\},\$4\rangle$, $\langle\{2\},\$5\rangle$, and $\langle\{1,2\},\$7\rangle$. Now the auctioneer can allocate items 1 and 2 to the bidder for \$9. To allow bidders to express any valuation $v : 2^M \rightarrow \{\Re_+ \cup 0\}$, it was proposed that bidders can submit XOR-constraints between bids [9]. If two bids are combined with an XOR-constraint, only one of them can win.[5] It turns out that in the continuous case, there is an inherent tradeoff between the full expressiveness of XOR-constraints and computational complexity (recall that in the binary case, CAWDP is $\mathcal{NP}$-complete even without XOR-constraints):

Theorem 4.4 *If* XOR*-constraints are allowed between bids, BCAWDP and CCAWDP (as well as BCRAWDP and CCRAWDP) are $\mathcal{NP}$-complete (with and without free disposal), even if prices are integer.*

PROOF. By Theorems 2.4 and 2.5, even the special case where the bids are all on singletons is $\mathcal{NP}$-complete. □

It follows that winner determination under the other fully expressive bidding languages that have been proposed for combinatorial auctions (which are generalizations of the XORS language) - OR-of-XORs [10] and XOR-of-ORs [5] - is $\mathcal{NP}$-complete even in the continuous case. The widely advocated idea of expressing mutual exclusion among bids via dummy items that the bids share [1; 5], does not lead to a fully expressive bidding language at all in the continuous case (because the dummy items may be partially allocated to different bids).

[5] In addition to allowing full expressiveness to the bidders, XOR-constraints can be a useful tool for the auctioneer. For example, they can be used to encode that rival bidders cannot both be winners (by inserting an XOR-constraint from each of the bidder's bids to each of the rival's bids).

The heart of the difficulty with the side constraints of this section is that they would require a bid to be "counted" even if it is accepted only partially. As the theorems of this section entail, such a counting device cannot be encoded into a linear program (of polynomial size) unless $\mathcal{P} = \mathcal{NP}$.

4.3 Side Constraint under which the Continuous and Binary Case Are Easy

As we show in this section, some side constraints restrict the space of feasible allocations so dramatically that the winner determination problem becomes easy even for the binary case. Currently we are not aware of any constraints in this class that would be of great practical interest, but the following artificial constraint serves as an existence proof.

Definition 14 (EXTREME-EQUALITY) *Each bid and ask has to be accepted to the same extent: $\forall j, x_j = x$.*

Theorem 4.5 *CMUCEWDP and BMUCEWDP are polytime solvable under EXTREME-EQUALITY (with and without free disposal).*

PROOF. The continuous case is directly solvable by linear programming. In the binary case, simply try accepting all offers ($x = 1$) and rejecting them ($x = 0$). □

5 Hybridizing Combinatorial and Multi-Attribute Market Designs

There are at least two reasons for introducing multi-attribute techniques into (combinatorial) markets. First, in a basic auction (or reverse auction or exchange), each item has to be completely specified. In many settings, this is overly restrictive. It would be more desirable to leave some of the parameters of the items open, so that each player could propose in her bids the most suitable parameter combinations for her, such as delivery date, quality, insurance, etc. (each player could also specify different parameter combinations in different bids). This would avoid the problem of having to enumerate alternative parameter combinations as separate items up front. Second, a bid from one bidder can be more desirable than the same bid from another bidder (e.g., due to historical data on timeliness and quality of different bidders).

Multi-attribute utility theory is a tool for handling tradeoffs between different attributes, and computerized implementations of it for automated negotiation have been developed over the last 15 years (see, for example, [14; 16]). Recently, several companies have been founded to commercialize that technology: `Frictionless Commerce`, `BizBots`, `Perfect`, etc. However, to our knowledge, multiattribute and combinatorial markets have not been hybridized in the literature so far. We propose a way to hybridize them so as to gain the advantages of both.

Consider a (combinatorial) market design such as the ones discussed in this paper so far. Let $\vec{a_j}$ be a vector of the additional (non-price) attributes. Some of the attributes can be specific to bid j while others might not (such as quality of a certain line of products). The vector can include attributes revealed by the bidder as well as attributes whose values the recipient gets from other sources such as historical performance databases. We re-weight the bid prices based on the additional attributes. The new price of any bid j is

$p'_j = f(p_j, \vec{a_j})$. The re-weighting function f could be determined by any party, but in most markets it would be set by the recipient of the bids before he receives the bids (or in some cases even after, but this would, in general, affect the bidders' incentives). We then run the winner determination in the (combinatorial) market using prices p'.

Theorem 5.1 *Whether or not p' is used (for some of the bids) in the objective, and whether or not p' is used (for some of the bids) in the side constraints, the $\mathcal{NP}$-completeness and polytime results of this paper still hold.*

PROOF. The polytime result hold because any of the p-based coefficients can be changed to p'-based coefficients in the linear program. The $\mathcal{NP}$-completeness results hold because the special case $f(p_j, \vec{a_j}) = p_j$ is $\mathcal{NP}$-complete. □

6 Conclusions and Future Research

In most real-world (electronic) marketplaces, there are other considerations besides maximizing immediate economic value. We presented a sound way of taking such considerations into account via side constraints and non-price attributes. Side constraints have a significant impact on the complexity of clearing the market. Budget constraints, a limit on the number of winners, and XOR-constraints make even noncombinatorial markets $\mathcal{NP}$-complete to clear. The latter two make markets $\mathcal{NP}$-complete to clear even if bids can be accepted partially. This is surprising since, as we showed, even combinatorial markets with a host of very similar side constraints can be cleared in polytime. An extreme equality constraint makes combinatorial markets polytime clearable even if bids have to be accepted entirely or not at all. Finally, we presented a way to take into account additional attributes using a bid re-weighting scheme, and proved that it does not change the complexity of clearing. All of the results hold for auctions as well as exchanges, with and without free disposal.

Future research includes analyzing the complexity entailed by other side constraints. We also hope to design search algorithms that perform well on average on $\mathcal{NP}$-complete clearing problems that include side constraints.

References

[1] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 548–553, Stockholm, Sweden, August 1999.

[2] Michael R Garey and David S Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[3] Holger Hoos and Craig Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 22–29, Austin, TX, August 2000.

[4] Daniel Lehmann, Lidian Ita O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 96–102, Denver, CO, November 1999.

[5] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 1–12, Minneapolis, MN, 2000.

[6] Michal Penn and Moshe Tennenholtz. Constrained multi-object auctions and b-matching. *Information Processing Letters*, 75(1–2):29–34, July 2000.

[7] Michael H Rothkopf. Bidding in simultaneous auctions with a constraint on exposure. *Operations Researche*, 25:620–629, 1977.

[8] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[9] Tuomas Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Proc. of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 542–547, Stockholm, Sweden, 1999. Extended version first appeared as Washington Univ., Computer Science, WUCS-99-01, Jan. 28th.

[10] Tuomas Sandholm. eMediator: A next generation electronic commerce server. In *Proceedings of the Fourth International Conference on Autonomous Agents (AGENTS)*, pages 73–96, Barcelona, Spain, June 2000. Early version appeared in the AAAI-99 Workshop on AI in Electronic Commerce, Orlando, FL, pp. 46–55, July 1999, and as a Washington University, St. Louis, Dept. of Computer Science WU-CS-99-02, Jan. 1999.

[11] Tuomas Sandholm and Subhash Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 90–97, Austin, TX, 2000.

[12] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, WA, 2001.

[13] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. Winner determination in combinatorial auction generalizations. In *AGENTS-01 Workshop on Agent-Based Approaches to B2B*, Montreal, Canada, 2001.

[14] K. Sycara. Utility theory in conflict resolution. *Annals of Operations Research*, 12:65–84, 1988.

[15] Moshe Tennenholtz. Some tractable combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX, August 2000.

[16] Gaurav Tewari and Pattie Maes. Design and implementation of an agent-based intermediary infrastructure for electronic markets. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, Minneapolis, MN, 2000.