Flexible double auctions for electronic commerce: theory and implementation

Peter R. Wurman [*], William E. Walsh [1], Michael P. Wellman [2]

Artificial Intelligence Laboratory, University of Michigan, 1101 Beal Ave., Ann Arbor, MI 48109-2110, USA

Abstract

We consider a general family of auction mechanisms that admit multiple buyers and sellers, and determine market-clearing prices. We analyze the economic incentives facing participants in such auctions, demonstrating that, under some conditions, it is possible to induce truthful revelation of values by buyers or sellers, but not both, and for single- but not multi-unit bids. We also perform a computational analysis of the auctioneer's task, exhibiting efficient algorithms for processing bids and calculating allocations. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Double auctions; Incentive compatibility; Electronic commerce mechanisms

1. Introduction

A commercial interaction is composed of at least three distinct steps. Potential buyers and sellers must first find one another, then negotiate the terms of the exchange, and finally, execute the transaction. Electronic commerce promises to automate part, or all, of each of these steps. Resource finding is facilitated by search engines and shopping agents [4], and electronic payment mechanisms [9] execute part of the exchange. Infrastructure for automated negotiation has not received as much attention, although the proliferation of online auctions on the World Wide Web [3] is evidence that explicit price formation mechanisms will play an important role in electronic commerce.

To support our research into negotiation protocols for electronic commerce and multiagent systems [20,21], we have built the Michigan Internet Auctionbot [23], a configurable auction server deployed over the World Wide Web (http://auction. eecs.umich.edu/). The AuctionBot has been operational since September 1996, allowing human agents to create auctions and submit bids via web forms, and (more recently) software agents to perform the same operations via TCP/IP. A user can configure

---

[*] Corresponding author. E-mail: pwurman@umich.edu
[1] E-mail: wew@umich.edu
[2] E-mail: wellman@umich.edu

[3] As of July 1998, Yahoo http://www.yahoo.com lists 95 online auctions on the web — and their list is far from complete. According to industry reports, two of these have gross sales approaching US$100 million/year, and online auction sales totalled US$500 million in 1997.

the AuctionBot to administer a variety of auction types, by setting parameters controlling the bidding protocol and auction rules. An auction description specifies such attributes as the number and frequency of market-clearing events, restrictions on bidding, revelation of intermediate information, and the policies for determining prices and matching buyers and sellers [22]. Options currently offered in the Auction-Bot cover most of the traditional auction types described in the literature, and we continue to extend the system to support a widening variety of auction mechanisms.

The flexibility offered by a configurable server is valuable not only for research and experimentation purposes, but also for the online service itself. Different negotiation mechanisms are appropriate in different circumstances, and so any generic mediation service should support a range of options. Fortunately, the concept of auction is general enough to accommodate a rich space of mechanisms.

Auctions also offer the advantage of *regularity*. Auction processes tend to operate via simple and well-structured interfaces, defined in terms of standardized languages for expressing bids, and for describing outcomes. This facilitates development of convenient user interfaces for human participants, and of software agents for automated negotiation.

A third advantage of working within the auction framework is that we can draw on a large body of theoretical work in auction theory and mechanism design. Auction theory analyzes strategies and outcomes of the auction as a game, yielding characterizations of equilibrium behavior and outcome properties. By formalizing the auction as a computational protocol as well, we can bring to bear standard techniques for algorithm design and complexity analysis. In some cases, we may encounter tradeoffs between desirable economic and computational properties, causing us to pursue principled ways to make such tradeoffs in practical negotiation settings.

In this paper, we examine a general family of auction mechanisms that admit multiple buyers and sellers, and determine market-clearing prices. We analyze the economic incentives facing participants in such auctions, demonstrating that it is possible to induce truthful revelation of values by single-unit buyers or sellers, but not both, and not for multi-unit bids. We also perform a computational analysis of

the auctioneer's task, exhibiting efficient algorithms for processing bids and calculating allocations.

2. Auction design

Most of the classic auctions examined in introductory surveys of auction theory [10,14] are *one-sided*, in that a single seller (or buyer) accepts bids from multiple buyers (or sellers). Two-sided, or *double* auctions [5], in contrast, permit multiple buyers and sellers to bid to exchange a designated commodity. The *continuous double auction* (CDA), matches buyers and sellers immediately on detection of compatible bids. A periodic version of the double auction (sometimes termed a *call market* [11] or *clearinghouse*) instead collects bids over a specified interval of time, then *clears* the market at the expiration of the bidding interval [4].

Most types of auctions can be characterized in terms of how they manage three core activities.

1. *Process bid*. The auction checks a bid for validity, and updates its database of active bids accordingly.

2. *Generate price quote*. During the bidding period, the auction may reveal information about the status of bids. We refer to such reports generically as *price quotes*, as they are typically defined in terms of hypothetical prices. In the AuctionBot and in this paper, we define a price quote as the price that the agent would have had to offer in order for its bid to have been accepted had the auction cleared at the time of the quote. This definition is expressed in past tense and subjunctive mood to emphasize that the price quote represents historical hypothetical information.

3. *Clear*. The primary function of an auction is to determine contracts between compatible buyers and sellers. The clear action matches buyers and sellers, and sets the transaction price.

---

[4] We refer to these double auctions as *periodic* to contrast them with the CDA, in which clears are triggered by the reception of new bids. Despite the name, the class of periodic double auctions includes those with aperiodic clear policies, such as those triggered by elapsed periods of inactivity, a randomly generated schedule, or some predefined clearing schedule.

For single-unit auctions, a bid simply indicates whether it is to buy or sell, and at what price. A multi-unit bid generalizes this by specifying a set of price–quantity pairs, where negative quantities correspond to offers to sell and positive quantities to buy. We generally require that quantity be decreasing in price-per-unit. An agent can make both buy and sell offers at the same time—the decreasing-in-price requirement ensures that the sell offers are at higher prices than the buy offers.

We generally assume that each agent may have only one active bid at a time—a new bid submission replaces any old bids. An agent can modify its current bid simply by submitting a new one. An agent can effectively withdraw its bid by submitting a null bid—one specifying zero quantity. However, we often find it useful to support bid withdrawal as a distinct activity.

Our analysis assumes that all multi-unit bids are *divisible*: an agent willing to buy q units at a specified price-per-unit would also be willing to exchange q', $0 \le q' < q$ at that price. Similarly, an agent willing to sell q' units at a specified price-per-unit would accept a transaction to exchange q', $0 \ge q' > q$ units at that price (recall that quantities for sellers are expressed as negative numbers). In other words, agents may not submit 'all or nothing' bids. In the discussion below, we sometimes invoke an assumption that each bid be for a single unit. For purposes of incentive analysis, this restriction is substantive. However, for describing our auction implementation it is strictly a matter of convenience, as we can translate any divisible multi-unit bid into an equivalent set of single-unit bids.

Price quotes, like clears, can be triggered in several ways. Our theoretical results in Section 4 are restricted to *sealed-bid* auctions, which clear exactly once, and do not generate price quotes. The algorithms discussed in Section 5 apply to all of the periodic price quote and clear policies.

If all exchanges mandated by the auction clearing policy are to occur at the same price, we call such a mechanism a *uniform-price* auction. One argument in favor of uniform-price mechanisms is their perceived fairness. A mechanism that generates multiple prices for the same good at the same time may be seen as unfair by the participants. A second benefit of a uniform-price mechanism is that it simplifies the auctioneer's task of calculating *price quotes*. If discriminatory prices are used, the auction may need to calculate a separate price quote for each individual.

The AuctionBot supports both one- and two-sided auctions (the former as a simple restriction on the latter), and both continuous and periodic double auctions.

3. Mth-price and $(M + 1)$st-price rules

Consider a set of L single-unit bids, of which M are sell offers and the remaining $N = L - M$ are buy offers. The Mth-Price auction clearing rule sets the price at the Mth highest among all L bids. Similarly, the $(M + 1)$st-price rule chooses the price of the $(M + 1)$ bid. Note that the Mth price is undefined if there are no sellers, and the $(M + 1)$st price is undefined if there are no buyers.

The Mth- and $(M + 1)$st-price rules use the same method for determining which bids belong in the transaction set. Let m denote the number of unit sell offers at or below the clearing price, and n the number of unit buy offers at or above the clearing price. Let $a = \min(m,n)$. The transaction set consists of the a highest unit buy offers and the a lowest unit sell offers. Auctions applying these rules may arbitrarily break ties and arbitrarily match successful buy and sell bids to each other.

The significance of the Mth and $(M + 1)$st prices follows from the fact that they determine the price range that balances supply and demand. Consider a simple case where there is one buyer willing to pay no more than $\$x$, and one seller willing to accept no less than $\$y$, with $x > y$. The Mth price is $\$x$, and the $(M + 1)$st price is $\$y$. If we set the price for the good above $\$x$, then one agent would be willing to sell it, but no agent would be willing to buy it. At a price below $\$y$, there is demand for one unit but no supply. Only if the price is between $\$y$ and $\$x$ is the excess demand zero.

We use this observation to define the price quote corresponding to these auction rules. The standard price information in a double auction is the *bid–ask quote*. In a CDA, the bid and ask quotes correspond to the highest unmatched buy offer and the lowest unmatched sell offer, respectively. However, it is the

information content of the price quote that is important: the bid quote is the price that a seller must offer in order to trade, and the ask quote reflects what a buyer must offer to trade. The generalization to our family of auctions is direct: the Mth price is the ask quote, and the $(M + 1)$st price is the bid quote. Notice that this definition generalizes the common notion of bid–ask spread, since it applies even if the buy and sell bids overlap.

In fact, the Mth- and $(M + 1)$st-price rules belong to a more general class of mechanisms called k-double auctions [17]. The parameter $k \in [0,1]$ specifies the fraction between the $(M + 1)$st and Mth prices at which the clearing price is set. The algorithm we present in Section 5 applies to the entire class of k-double auctions. However, our analysis in the next section is concerned only with the two extreme points, $k \in \{0,1\}$, of the equilibrium price range.

The price quote reveals to agents salient information bearing on whether their current bids would be in the transaction set. Clearly, if its buy offer is above the ask quote, or its sell offer is below the bid quote, the agent has a winning bid. If an agent's buy offer is equal to the ask quote, and the ask quote is strictly greater than the bid quote, then the agent's bid is winning. This follows from the fact that the ask quote is the Mth highest bid. Let b_M be the buy bid at the Mth highest price. Suppose there are a total of x buy bids at or above b_M. If the $(M + 1)$st highest bid is strictly less than b_M, there must be $M - x$ sell bids at or above b_M, and therefore $M - (M - x) = x)$ sell bids at or below the $(M + 1)$st bid. Therefore b_M is in the current match set. Similarly, if a seller's offer equals the bid quote and is strictly less than the ask quote, its bid is winning.

However, if the bid and ask quotes are equal, neither buyers nor sellers with offers at that price can determine whether they are winning or losing. In both examples in Table 1, the Mth price $= (M + 1)$st price $= \$2$. In Example 1a, one buy offer would be unmatched if the auction cleared. In Example 1b, one sell offer would be left unmatched. Although the price setting rules will not leave *both* buy and sell bids unmatched at the clearing price, it is not possible to tell from the price quote which side is guaranteed to clear completely and which will be subject to the tie-breaking criteria.

Table 1
Examples in which (a) a buy bid is unmatched at the Mth price, and (b) a sell bid is unmatched at the $(M + 1)$st price

(a)	(b)
Buy 1 unit at $2	Buy 1 unit at $2
Buy 1 unit at $2	Sell 1 unit at $2
Sell 1 unit at $2	Sell 1 unit at $2

When $M = 1$, the Mth price is simply the highest buy offer (assuming it is greater than the seller's reservation price), and the $(M + 1)$st price corresponds to the second highest. The highest, or first, price expressly offered is used by the classic English, Dutch, and first-price sealed-bid auctions [10,14]. The second-price rule was proposed and analyzed by Vickrey in his seminal work establishing the field of auction theory [19]. [5] When $M > 1$, the Mth- and $(M + 1)$st-price auctions correspond to the multi-unit generalizations of Dutch and English auctions studied by McCabe et al. [11].

4. Incentive compatibility analysis

The Mth- and $(M + 1)$st-price rules are symmetric with respect to their treatment of buyers and sellers. We therefore present the proof of our incentive compatibility result below in detail for the $(M + 1)$st-price, sealed-bid auction, and leave it to the reader to make the necessary minor adjustments to get the analogous result for the MthPrice, sealed-bid auction. Note that rather than using M—the number of units offered for sale—we could define the rule in terms of N, the number of unit buy offers. The pricing rule in which we count up N units from the lowest bid is identical to the $(M + 1)$st-price rule, and the $(N + 1)$st-price rule is identical to the Mth.

---

[5] In our AuctionBot implementation, an English auction uses the Mth-price rule, issues price quotes after every bid, and clears once based on a period of bidder inactivity. Despite the fact that the English auction uses the highest-mentioned buy offer, under the independent private values model (discussed in Section 4), its equilibrium outcome is identical to the second-price auction, since buyers bid only in increments from their counterparts. There is a corresponding equivalence between the outcomes, and the strategic problems faced by agents, in the first-price sealed-bid and Dutch auctions.

In the following theoretical analysis, we assume that agents have *independent private values* for the goods, which means that each agent knows its own valuation, which is unaffected by the value other agents place on the good [10].

An auction is *incentive compatible* if the agents optimize their expected utilities by bidding their true valuations for the good. This is a desirable feature because an agent's decision depends only on its local information, and it gains no advantage by expending effort to model other agents. It is important to note that the concept of incentive compatibility is meaningful only under the independent private values assumption. If the agents have uncertain correlated values, then, in any of the auctions considered here, an agent should generally bid below its estimated valuation in order to avoid the 'winner's curse' [13].

An auction is *individually rational* if its allocations do not make any agent worse off than had the agent not participated.

An allocation is *efficient* if there are no further gains from trade possible. This implies that the goods are allocated to the agents who value them most highly.

Uniform-price auctions tend to promote efficiency when there are multiple buyers and sellers. Consider four truthful single-unit bids: two offers to sell, for $x and $z, and two offers to buy, for $w and $y, where $w > x > y > z$. Let W, X, Y, and Z refer to the agents who bid $w, $x, $y, and $z, respectively. There are two obvious candidate allocations. In the first, W and Z trade. This trade could be supported by a uniform-price rule that sets the price between $x and $y. Most importantly, this trade allocates the goods to the agents who value them most highly—agent W buys one and agent X keeps one. In the second allocation, W trades with X, and Y trades with Z. These trades would necessarily occur at two different prices, and the agents who end up with the goods have the first and third highest valuations. This allocation is inefficient because the agents X and Y could both be made better off by a further trade between them. Although the second allocation is an improvement over the initial endowments, it is dominated by the uniform-price allocation.

As noted above, for $M = 1$ the $(M + 1)$st-price auction is the well-studied second-price mechanism. Vickrey [19] showed that the sealed-bid version of this mechanism is incentive compatible for the buyers. The intuition behind this result is as follows. Recall that the agent with the highest buy offer wins the good but pays the price of the second highest offer. Clearly, bidding more than the good's true value is irrational because it exposes the agent to a potentially unprofitable transaction while not increasing the likelihood of a profitable one. By bidding less than its true value, the agent decreases the probability that it wins the bid, but does not change the amount it pays if it wins. Both strategies make the buyer worse off than bidding truthfully.

We can generalize this result to single-unit buyers in the $(M + 1)$st-price, sealed-bid auction, allowing multiple sellers as well as other agents (buyers or sellers) with multi-unit (divisible) bids.

Theorem 1. The $(M + 1)$st-price sealed-bid auction is incentive compatible for single-unit buyers under the independent private values model.

Proof. Let us define a full ordering of the bids as one where they are in decreasing order by price, and where multi-unit bids are treated as independent single-unit offers. The clearing price is determined by the $(M + 1)$st order statistic, b_{M+1}, of the bids. Let i be a single-unit buyer whose independent valuation for one unit of the good is v_i and whose bid is b. Buyer i's expected utility is the product of the probability of winning a unit of the good and the utility gained from winning:

$$\left[\Pr(b > b_{M+1}) + \Pr(b = b_{M+1})\Pr(\text{win}|b = b_{M+1})\right]$$
$$\times (v_i - b_{M+1}).$$

If $v_i - b_{M+1}$ is positive, the buyer wants to maximize the probability of winning. Because $v_i > b_{M+1}$, the probability of winning is 1 when $b = v_i$. If $v_i - b_{M+1}$ is negative, the buyer wants to minimize the probability of winning. In this case, the probability of winning is zero if $b = v_i$. It follows that setting $b = v_i$ is always an optimal strategy. Moreover, truth telling is a *dominant strategy* because it is optimal regardless of the other agents' strategies. □

It might be hoped that the $(M + 1)$st-price auction would also be incentive compatible for single-unit sellers. Alas, this is not the case. The reason is that if the $(M + 1)$st price falls on a seller's bid, that seller

can increase its profits by increasing its bid up to the Mth price. Thus,the seller does not maximize its utility by being truthful. As an illustration, consider the simple case where a single buyer values the good at $10 and a single seller values the good at $0. If the auction were incentive compatible for sellers, then the seller would maximize its utility with a bid of $0. If the seller did so, it would receive the second-price payment of $0. Clearly the seller could do better by bidding as high as $10—selling the good for a higher price. In contrast, the buyer could not reduce the amount it pays by reducing its own bid.

The symmetry between the Mth and $(M + 1)$st rules yields a counterpart of Theorem 1 for the Mth-price auction.

Theorem 2. The MthPrice sealed-bid auction is incentive compatible for single-unit sellers under the independent-private-values model.

Proof. The proof is identical to that of Theorem 1, with appropriate reversals of inequalities and substitutions of sellers for buyers, and Mth for $(M + 1)$st. □

The $(M + 1)$st and Mth-price auctions offer a choice between incentive compatibility for buyers or for sellers. The only way to get incentive compatibility for both is if some party is willing to subsidize the auction. Myerson and Satterthwaite [15] show that there does not exist any bargaining mechanism that is individually rational, efficient, and Bayesian incentive compatible for both buyers and sellers, that does not require outside subsidies.

We next consider another relaxation of the theorems' conditions, specifically the restriction to single-unit bids. The generalized Vickrey auction (GVA) [8,18]—an extension of mechanisms developed by Vickrey [19], Clarke [2], and Groves [7]—is incentive compatible for all bidders. [6] The GVA is a *direct revelation* mechanism [6]; agents submit their reservation prices (equivalent to their utility func-

tions under the independent private values model) as bids. The GVA calculates an efficient allocation of the goods, and net payments for each agent. Agent j's payment is its impact on the social welfare: the value that the other agents would achieve if j were not present, minus the value that they obtain with j included.

However, the GVA, even when restricted to a single-sided auction, does not produce a uniform price. Consider an example with two buyers and two identical units of a good. Agent A values one unit of the good at $5, and two units at $7. Agent B values one and two units at $5 and $8, respectively. The efficient allocation is to give one unit to each agent. However, the GVA payment for agent A is $8 − $5 = $3, whereas the payment by agent B, for a unit of the same good, is $2. Thus, the GVA mechanism does not satisfy our goal of a uniform-price, incentive compatible mechanism for buyers who desire multiple units.

The $(M + 1)$ $(M$thPrice$)$ auction computes a uniform price, but is not incentive compatible for multi-unit buyers (sellers). Consider an $(M + 1)$st-price auction in which a seller offers one unit of a good for $1 and the second unit for $10. There is a single buyer with reserve prices $7 and $5 for the first and second units of the good, respectively. If the buyer bids its true values, then the $(M + 1)$st-price is its bid for the second unit, and it will receive one unit for $5. However, if the buyer lowers its bid for the second unit to $1, it would receive one unit for $1. The common principle behind the incentive results for the GVA and those in Theorems 1–2 is that an agent's payment is a function of other agents' bids, but not of its own. When we admit multi-unit bids in the $(M + 1)$st-price auction, the price that the buyer pays for one unit may be set by its bid for the other unit, creating a built-in collusion [19]. As a result, it is suboptimal for the buyer to place truthful multi-unit bids. We now show that this result generalizes to a broad class of uniform-price auctions.

---

[6] To ensure incentive compatibility for both buyers and sellers in the GVA, it may be necessary to subsidize the mechanism. For the purposes of finding an incentive compatible mechanism for multi-unit bids, we restrict the discussion to only single-sided GVA mechanisms.

Theorem 3. There does not exist an individually rational, uniform-price auction with multi-unit allocations that is Bayes−Nash incentive compatible for buyers or sellers, and guaranteed to produce efficient allocations.

Proof. We consider the case of incentive compatibility for buyers only. By symmetry, the theorem extends to the case of incentive compatibility for sellers only.

It is sufficient to show that there exists a situation for which the properties do not all hold. Assume there are two units of a good, and two buyers, denoted $i = 1$, 2. Each buyer's utility can be described by reservation prices. Buyer i's marginal utility for the jth unit of the good is v_i^j. Assume that marginal utility is nonincreasing with j. Let r_i be the bid reported by agent i, and $p(r1, r2)$ the price determined by the auction as a function of the bids.

By the *revelation principle* [6], we can restrict our attention to a mechanism in which the buyers report their marginal valuations for each unit of the good. Buyer i sends a report $r_i = (r_i^1, r_i^2)$, where r_i^j is its reported marginal utility for the jth unit of the good.

If the auction is efficient, it must allocate k units of the good to an agent if the agent reported k of the two highest marginal valuations. If it is individually rational and incentive compatible, no agent pays more than its reported value for an allocation.

Consider the situation where the marginal valuations are ordered $v_1^2 < v_2^2 < v_2^1 < v_1^1$. Assume that buyer 1 reports its true valuations, but buyer 2 reports $r_2 \neq v_2$, such that

$$r_1^2 < r_2^2 < r_2^1 < r_1^1. \tag{1}$$

If the auction is efficient, each agent receives one unit of the good. Incentive compatibility entails that $p(v_1, r_2) \geq p(v_1, v_2)$, otherwise buyer 2 would not optimize by reporting truthfully. But because this must be true for any r_2 and v_2 that obey the inequalities, and because the auction receives only the reports, we must have $p(v_1, r_2) = p(v_1, v_2)$ for any such values and reports.

This further implies that the price is arbitrarily close to the lowest reported value,

$$p(r_1, r_2) \leq r_1^2 + \epsilon \tag{2}$$

for any $\epsilon > 0$.

Now consider the situation where marginal valuations are ordered $0 < v_1^2 < v_1^1 < v_2^2 < v_2^1$, and $v_1^1 = v_2^1 + \sigma$. If both buyers report their true valuations, then buyer 1 receives nothing (and thus zero gain in utility) from the auction allocation. But if buyer 2 were to report truthfully and buyer 1 were to report

$r_1^2 = v_1^2$ and $r_1^1 > v_2^1$, then Eq. (1) holds and buyer 1 would win one unit. Because Eq. (2) holds for $0 < \epsilon < \sigma$, it follows that $p(r_1, r_2) < v_1^1$, giving buyer 1 a positive gain in utility. This violates incentive compatibility, revealing a contradiction among the posited auction properties. □

Although the incentive compatibility properties of the Mth- and $(M + 1)$st-price sealed-bid auctions are somewhat restricted, we see from Theorems 1–3 and the impossibility result from Myerson and Satterthwaite that we can extend their scope only by compromising some other desirable properties. Thus when we consider it important to have a single price for a commodity, it may well make sense to choose a mechanism from this family.

5. Implementation

A great deal of work in auction theory has focused on the economic efficiency of various mechanisms. Relatively little research has focused on their computational efficiency. Whereas it is true that in most cases the costs of communication outweigh the costs of managing the auction, it is still useful to define efficient algorithms for the mediators. This is especially relevant to auctions covering multiple commodities, in which clearing prices are often determined by combinatorial optimization. Rothkopf et al. [16] study restrictions in allowable multicommodity bundles that enable tractable solution of the auction's optimization problem.

Even when each auction faces a tractable problem, we might obtain significant computational savings by careful management of the communication and clearing operations. Andersson and Ygge [1] apply hierarchical market structures for distributing the calculation of market-clearing prices and allocations. Their CoTree algorithm allows designers to make tradeoffs between the number of messages, and the dimensionality of local excess demand calculations.

In this section, we present an efficient algorithm for the Mth- and $(M + 1)$st-price family of auctions. Our analysis reveals tradeoffs between bid-processing methods based on the expected pattern of bid updates and clearing events.

5.1. Operations

To support the core activities discussed in Section 2, an auction algorithm must implement the following operations:

Insert/Remove: place a new bid into the data structure, or remove a current one.

Price Quote: calculate the bid and ask quotes given the current set of bids.

Clear and Match: calculate a clearing price and remove all of the bids that match.

Note that when an agent modifies its bid, the auction can implement this by removing the agent's current bid and inserting a new one.

Perhaps the most straightforward implementation of the rules of Section 3 would be to maintain a sorted list of all bids, and perform clears as described. Assume that we have $L = M + N$ single-unit bids in a sorted list. Inserting a new bid takes $O(L)$ time. Generating a price quote takes $O(M)$ time. Removing a bid can be accomplished in constant time if a secondary mechanism (such as a hash table) is used to associate bidders with their bids. Clearing and matching is an $O(L)$ operation because we have to trace through all of the bids to find the ones that matched. We could do somewhat better—clearing in $O(\min(M,N))$—by employing two lists, one for buy bids and one for sells. In either case, when the number of bids is small, a sorted-list algorithm might be fine. We can, however, do better for large M and N.

5.2. The 4-heap algorithm

Our algorithm uses four heap structures to organize the bids. We distinguish bids by whether they represent buy or sell offers, and whether or not they are in the current match set. The four heaps are:

B_{in}: Contains all of the buy bids that are in the current match set. The heap priority is minimal price, so that the lowest priced bid is on top. The size of this heap is $O(\min(M,N))$.

B_{out}: Contains all of the buy bids that are *not* in the current match set. The heap priority is maximal price. The size of this heap is $O(N)$.

S_{in}: Contains all of the sell bids in the match set, prioritized by maximal price. The size of this heap is $O(\min(M,N))$.
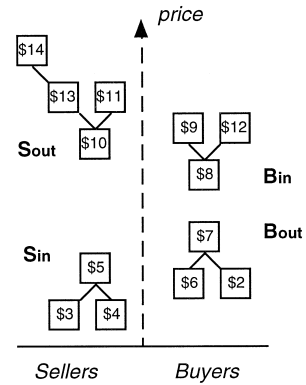

Fig. 1. Schematic diagram of bids arranged in the four heaps.

S_{out}: Contains all of the sell bids not in the match set, prioritized by minimal price. The size of this heap is $O(M)$.

Fig. 1 illustrates the relationships between the heaps.

Recall that a heap data structure is a complete binary tree, with the property that each node has a priority not exceeding its parent's [3]. Inserting a new node into the heap (put), or removing the top node (get) each take $O(\log K)$ time, where K is the size of the heap.

The 4-HEAP algorithm ensures several constraints among the heaps. First, the number of units in B_{in} must equal the number in S_{in}. The rest of the constraints are characterized in terms of the top nodes of each heap. Let b_{in}, b_{out}, s_{in}, and s_{out} be the top nodes of B_{in}, B_{out}, S_{in}, and S_{out}, respectively. Value(n) is the value of node n. We enforce the following constraints:

- $\text{Value}(b_{in}) \geq \text{Value}(b_{out})$
- $\text{Value}(s_{out}) \geq \text{Value}(s_{in})$
- $\text{Value}(s_{out}) \geq \text{Value}(b_{out})$
- $\text{Value}(b_{in}) \geq \text{Value}(s_{in})$

McCabe and Smith [12] also study the design of uniform-price, double auctions, and organize their bids into four sets, although they use sorted lists rather than heaps.

5.3. Complexity analysis

The complexity of bookkeeping in the 4-HEAP algorithm comes from the need to keep B_{in} and S_{in} the same size while performing inserts and removes.

To simplify the description, we restrict attention to single-unit bids. The extension to multi-unit bids is relatively straightforward, and is discussed in Section 5.4.

5.3.1. Insert

When a new bid comes in, the algorithm may need to do more than place it in one of the heaps. Using the example depicted in Fig. 1, if the auction receives a sell offer for $6, not only would it be placed onto S_{in}, but the top bid in B_{out} must be transferred into B_{in} to equilibrate the *in* heaps. This requires one get from Bout, one put into S_{in} and one put into B_{in}. Receiving a buy bid also requires as many as three heap operations. Thus, the insert operation for an arbitrary bid is bounded by $O(\log L)$.

In general, when a new sell bid, s_{new}, arrives, there are three possible actions. Either the new bid forms a new match with the top bid in B_{out}, or it displaces a bid in S_{in}, or it is placed into S_{out}. The pseudocode is shown in Fig. 2.

The logic for new buy bids is very similar, with all of the Bs and Ss switched, and the inequalities reversed.

5.3.2. Remove

To remove bids efficiently, we employ an external lookup mechanism, such as a hash table, to locate the bid within its containing heap in constant time. We can then delete this node from its heap in logarithmic time. If the bid is in one of the *out* heaps, we are finished. However, if it is in S_{in} or B_{in}, we must also transfer the top bid from the other

in heap to its corresponding *out* heap. Thus, in the worst case, removing a bid requires three heap operations, for a total time bounded by $O(\log L)$.

5.3.3. Clears and quotes

A price quote can be generated simply by inspecting the tops of the heaps. The bid quote, or $(M + 1)$st price, is max(Value(s_{in}), Value(b_{out})). The ask quote, or Mth price, is max(Value(s_{out}), Value(b_{in})). The constraints on the heaps ensure that the ask quote is above the bid quote. Calculating a price quote is a constant time operation.

Clearing prices for the Mth-price rule are set to the ask price. The $(M + 1)$st auction clears at the bid price. We match buyers and sellers by disassembling the *in* heaps, B_{in} and S_{in}. Matching necessarily takes time proportional to the number of bids matched, which is $O(\min(M,N))$.

5.4. Multiple-unit auctions

The 4-HEAP algorithm can be extended to allow nodes to represent multi-unit bids, with a relatively simple modification. Bids transferred between *in* and *out* heaps may need to be split in order to maintain an exact equivalence in the number of units stored in the two *in* heaps. Similarly, insertion or removal of a multi-unit bid may entail several nodes be transferred across the complementary pair of *in* and *out* heaps.

A simple example illustrates the point. Agent A submits a bid to sell two units at $3. The auction places this bid into S_{out}. Agent B now submits a bid to buy one unit at $5. The auction puts the new bid

```
if (( Value(s_new) ≤ Value(b_out)) and ( Value(s_in) ≤ Value(b_out)))
        put(s_new, S_in)
        b ← get(B_out)
        put(b, B_in)
else if ( Value(s_new) < Value(s_in))
        s ← get(S_in)
        put(s, S_out)
        put(s_new, S_in)
else
        put(s_new, S_out)
```

Fig. 2. Pseudocode for receiving a new sell bid.

into B_{in}, and splits the first bid, moving one unit into S_{in} and leaving one unit in S_{out}. When the auction receives a third bid, from agent C to buy two units at $4, it moves the remaining unit of the first bid from S_{out} to S_{in}, and splits the third bid between B_{in} and B_{out}.

It is easy to see that any multi-unit bid may, over time, be split entirely into single-unit bids. Hence, the worst-case complexity of bid operations in the 4-HEAP algorithm must be characterized in terms of number of units, rather than number of bids.

5.5. Tradeoffs between operation costs

Based on the analysis of operations above, we see that the 4-HEAP algorithm can process bids (insert and remove) in $O(\log L)$ time, issue price quotes in $O(1)$ (constant) time, and perform clears in $O(\min(M,N))$ time.

As a benchmark, we introduce the *simple sealed-bid* algorithm, which makes a different tradeoff between the computational costs of the respective operations. This algorithm works as follows. As each bid is received, append it to one of two unordered lists, representing the buy and sell bids. This is a constant time operation. To clear the auction, sort the buy offers in descending order by price, and the sell offers in ascending order. While the top buy offer is greater than or equal to the top sell offer, remove the front element of each list and place them in the transaction set. The computational complexity of this algorithm is in terms of R, the total number of price points in the bids (in contrast to the number of units). For instance, an agent who bids to sell two units at $5 and two at $10, has submitted one bid which has two price points and offers four units. The clear is an $O(R \log R)$ operation. A price quote is the same as a clear, except that the original lists are not destroyed.

When bids are restricted to single units, $R = L$, and both the 4-HEAP and simple sealed-bid algorithms require $O(L \log L)$ total time over the auction's life cycle, assuming a constant number of quotes and clears, and a constant number of bid revisions per unit bid. The simple sealed-bid algorithm is most appropriate for single-clear, sealed-bid auctions, since price quotes are relatively expensive. The 4-HEAP algorithm offers reduced clear-time latency at the expense of more work during the bid processing stage, and will be clearly superior in settings demanding frequent price quotes. We expect these observations hold for reasonably large sized bids and moderate bidding activity.

However, when there is a large variation in the sizes of bids, and a high frequency of withdraws and edits, it is possible that the simple sealed-bid algorithm will outperform the 4-HEAP algorithm even in auctions with frequent price quotes and multiple clears.

6. Conclusion

This paper contributes to both the theoretical and practical understanding of a useful family of periodic double auction mechanisms. We show that the Mth-Price sealed-bid auction is incentive compatible for single-unit sellers, and that the $(M+1)$st-price sealed-bid auction is incentive compatible for single-unit buyers. We also show, through a combination of our own results and those in the literature, that the incentive compatibility result cannot be extended to multi-unit bids, or simultaneously to both buyers and sellers.

We also present an algorithm that processes bids incrementally in order to reduce the time necessary for auction clears and price quotes. Bid processing requires logarithmic time, price quotes constant time, and clears linear time. Depending on the number and sizes of bids expected, and the relative frequencies of the various auction operations, this algorithm may offer advantages over the more straightforward approach.

Auctions have already begun to play an important role in electronic commerce. As automation of online negotiation becomes more widespread, we expect that developers will continue to introduce innovative auction mechanisms, and to apply them in novel commerce settings. Careful attention to both the economic incentive properties and computational requirements of auction mechanisms will undoubtedly be an important ingredient of their success.

tion Survivability program, and an NSF National Young Investigator award.

References

[1] A. Andersson, F. Ygge. Managing large scale computational markets, in: 31st Hawaiian International Conference on System Sciences, 1998, pp. 4–13.

[2] E.H. Clarke, Multipart pricing of public goods, Public Choice 11 (1971) 17–33.

[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT press, 1990.

[4] R.B. Doorenbos, O. Etzioni, D.S. Weld, A scalable comparison-shopping agent for the world-wide web. In First International Conference on Autonomous Agents, 1997, pp. 39–48.

[5] D. Friedman, J. Rust, (Eds.), The Double Auction Market: Institutions, Theories and Evidence, Addison-Wesley Publishing, Reading, MA, 1993.

[6] D. Fudenberg, J. Tirole, Game Theory, MIT press, 1991.

[7] T. Groves, Incentives in teams, Econometrica 41 (1973) 617–631.

[8] J.K. MacKie-Mason, H.R. Varian, Generalized Vickrey Auctions, Technical report, University of Michigan, July 1994.

[9] J.K. MacKie-Mason, K. White, Evaluating and selecting digital payment mechanisms, in: G.L. Rosston, D. Waterman (Eds.), Interconnection and the Internet, Lawrence Erlbaum, 1997.

[10] R.P. McAfee, J. McMillan, Auctions and bidding, Journal of Economic Literature 25 (1987) 699–738.

[11] K.A. McCabe, S.J. Rassenti, V.L. Smith, Auction institutional design: Theory and behavior of simultaneous multiple-unit generalizations of the Dutch and English auctions, American Economic Review 80 (5) (1990) 1276–1283.

[12] K.A. McCabe, V.L. Smith, Designing a uniform-price double auction: An experimental evaluation, in: Friedman and Rust (Eds.), The Double Auction Market: Institutions, Theories and Evidence, Addison-Wesley Publishing, Reading, MA, 1993, pp. 307–332.

[13] P. Milgrom, Auctions and bidding: A primer, Journal of Economic Perspectives 3 (3) (1989) 3–22.

[14] P.R. Milgrom, R.J. Weber, A theory of auctions and competitive bidding, Econometrica 50 (1982) 1089–1122.

[15] R.B. Myerson, M.A. Satterthwaite, Efficient mechanisms for bilateral trade, Journal of Economic Theory 29 (1983) 265–281.

[16] M.H. Rothkopf, A. Pekeč, R.M. Harstad, Computationally manageable combinational auctions, to appear in Management Science.

[17] M.A. Satterthwaite, S.R. Williams, Bilateral trade with the sealed bid *k*-double auction: Existence and efficiency, Journal of Economic Theory 48 (1989) 107–133.

[18] H.R. Varian, Economic mechanism design for computerized agents, in: First USENIX Workshop on Electronic Commerce, New York, 1995, pp. 13–21.

[19] W. Vickrey, Counterspeculation, auctions and sealed tenders, Journal of Finance 16 (1961) 8–37.

[20] W.E. Walsh, M.P. Wellman, A market protocol for distributed task allocation, in: Third International Conference on Multiagent Systems, Paris, 1998, pp. 325–332.

[21] W.E. Walsh, M.P. Wellman, P.R. Wurman, J.K. MacKie-Mason, Some economics of market-based distributed scheduling, in: 18th International Conference on Distributed Computing Systems, Amsterdam, 1998, pp. 612–621.

[22] P.R. Wurman, W.E. Walsh, M.P. Wellman, A parameterization of the auction design space, May 1998, submitted for publication.

[23] P.R. Wurman, M.P. Wellman, W.E. Walsh, The Michigan Internet AuctionBot: a configurable auction server for human and software agents, in: Second International Conference on Autonomous Agents, Minneapolis, 1998, pp. 301–308.
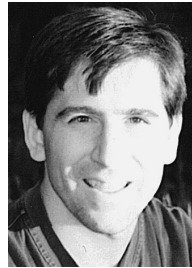
Peter Wurman is a PhD candidate in Computer Science and Engineering at the University of Michigan. He has MS degrees from the University of Michigan in Computer Science and Mechanical Engineering, and an SB degree from the Massachusetts Institute of Technology in Mechanical Engineering. Research interests include multiagent systems, market-oriented programming, and electronic commerce. Mr. Wurman is the recipient of the IBM Institute for Advanced Commerce's 1998 fellowship for Best Thesis Proposal in E-Commerce.

William E. Walsh is a PhD candidate in Computer Science and Engineering at the University of Michigan. He received an MS in Computer Science and Engineering, and a BSE in Computer Engineering, both from the University of Michigan. Research interests include computational market systems for decentralized resource allocation and electronic commerce. Mr. Walsh is a NASA/JPL graduate student researcher.

Michael P. Wellman is an associate professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. He received a PhD in Computer Science from the Massachusetts Institute of Technology in 1988 for his work in qualitative probabilistic reasoning and decision-theoretic planning. From 1988 to 1992, Dr. Wellman conducted research in these areas at the USAF's Wright Laboratory. Current research focuses on computational market mechanisms for distributed decision making, and electronic commerce. He is an NSF national young investigator, an AAAI councilor, and executive editor of the Journal of Artificial Intelligence Research.