Games and Economic Behavior **35,** 304–338 (2001)
doi:10.1006/game.2000.0828, available online at http://www.idealibrary.com on IDEAL®

A Parametrization of the Auction Design Space

Peter R. Wurman

Department of Computer Science, North Carolina State University, Raleigh, North Carolina 27695-7535
E-mail: wurman@csc.ncsu.edu

and

Michael P. Wellman and William E. Walsh

Computer Science and Engineering, University of Michigan, Ann Arbor, Michigan 48109-2110
E-mail: wellman@umich.edu; wew@umich.edu

Received May 19, 1998

We present an extensive breakdown of the auction design space that captures the essential similarities and differences of many auction mechanisms in a format more descriptive and useful than simple taxonomies. This parametrization serves as an organizational framework in which to classify work within the field and uncovers parameter combinations corresponding to novel mechanisms. The structured characterization of auction rules can be exploited for the modular design of configurable auction servers. It also facilitates the communication of auction rules to software agents, enabling the automation of flexible market-based negotiation. *Journal of Economic Literature* Classification Numbers: C70, D44.

1. INTRODUCTION

The manifest popularity of Internet auctions[1] suggests that automated auctions will play a significant role in electronic commerce. Indeed, this might be the case by definition, as any well-defined set of rules for determining the terms of an exchange of something for money can reasonably be characterized as an auction (McAfee and McMillan, 1987). Since

[1] Forrester Research forecasts that Internet-enabled business-to-business transactions will reach $406.2 billion in 2000, and jump to nearly $2.7 trillion in 2004. Much of this commerce will occur in electronic auctions (Blackmon, 2000). In fact, there were an estimated 1000 electronic “exchanges” online in early 1999, and that number is expected to grow to 10,000 within a year (Dalton, 1999).

0899-8256/01 $35.00

the automation of negotiation processes invariably requires their precise specification (in programs at least), the task of designing negotiation rules is essentially that of designing auctions. Thus, auction design constitutes a central activity for any situation where agents—humans, or software proxies—negotiate the exchange of resources. We have argued elsewhere (Wellman and Wurman; 1998a) that such situations will be commonplace for artificial agents, just as they are for humans engaged in electronic commerce.

The literature on auctions identifies a wide variety of auction types, which we discuss below. Despite the large number of online auctions,[2] the vast majority implement a variant of the English open-outcry auction, the type most familiar to the general public (Ungar *et al*., 1998; Wellman and Wurman, 1998b). Nevertheless, the rules and interfaces are not standardized, and so users must learn the conventions operating at a particular site. Those wishing to implement automated bidding agents must likewise customize their agents' behaviors to the individual sites. This makes it particularly difficult to coordinate bidding behavior across several sites, which may be desirable when identical items or close substitutes are auctioned simultaneously at multiple auction sites.[3]

To facilitate the development of agents able to participate in multiple auctions, we could standardize auction rules, or provide a standard way to *describe* auction rules. The latter approach is of course far more flexible and is what we advocate. Auctions would publish their rules in standard descriptive terms, interpretable by humans or software agents. Based on these descriptions, agents synthesize operational bidding strategies serving their objectives.

We have pursued this approach in the design of our configurable auction server: the Michigan Internet AuctionBot (Wurman *et al*., 1998b).[4] The AuctionBot implements a wide variety of auction mechanisms, as specified via the settings of a collection of orthogonal parameters. There is a natural correspondence between the decomposition of auctions into functional elements and an object-oriented approach to implementation.

[2] As of July 1999, Auction Watch (www.auctionwatch.com) lists over 300 consumer-oriented online auction sites.

[3] Emerging cross-auction services, including AuctionRover (www.auctionrover.com) and BiddersEdge (www.biddersedge.com), are search engines that combine the listings of dozens of online auctions. These sites are quickly adding features to support the tasks faced by both buyers and sellers. We expect that "shopping agents"—programs that search for products across multiple vendor sites (examples include mySimon (www.mysimon.com), Jango (Doorenbos *et al*., 1997) (www.jango.com), and Amazon's "Shop the Web" (shoptheweb.amazon.com))—will eventually be extended to support interactions in auctions (Guttman *et al*., 1998).

[4] auction.eecs.umich.edu

We have successfully exploited this similarity in our development of the AuctionBot—the majority of auction rules are encoded in a manner that allows them to be shared among the basic auctioneer programs.

In addition to its practical role in auction implementation and rule description, we believe that the parametrization is also useful in its own right as a characterization of the auction design space. Although the characterization as presented here is far from complete, we believe it a helpful start in organizing research on automated negotiation mechanisms.

The parametrization reported in this paper extends that currently implemented in the AuctionBot in two significant ways: we define the parameters for both linear and nonlinear pricing, and we do so in the context of multidimensional auctions—auctions that mediate the allocation of more than one resource type. Examples of these auctions include the FCC's Simultaneous Ascending Auction (McAfee and McMillan, 1996; McMillan, 1994), in which the eligibility rules create a loose coupling between resources, as well as expressly combinatorial auctions, which we discuss in more detail in Section 4.4.

We present the parametrization in several stages. Section 2 provides general definitions of the features common to all auctions, including a precise specification of bid semantics. In Section 3 we present the auction parameters and the values they can take. In Section 4 we discuss in more detail one particularly important parameter category: the policy by which an auction forms exchanges between agents. In Section 5 we classify some of the well-known auction types using our description language.

2. COMMON AUCTION CHARACTERISTICS

Many different types of auctions are in common use. The English open-outcry auction is often used to sell art and other collectibles, for example. The Dutch auction is commonly used to sell perishables, such as fish or flowers. First-price sealed bid (FPSB) and second-price sealed bid (SPSB, or *Vickrey*) auctions are most often used in procurement situations. Call markets and continuous double auctions (CDAs) are favored institutions for trading securities and financial instruments. These institutions and others are discussed in various auction survey papers (Engelbrecht-Wiggans, 1980, Friedman, 1993; McAfee and McMillan, 1987; Milgrom, 1987; Milgrom and Weber, 1982).

Some authors organize auction designs in a hierarchical taxonomy (Engelbrecht-Wiggans, 1980; Friedman, 1993). The straw-man example depicted in Fig. 1 classifies the auctions mentioned in the preceding paragraph by whether they are single or double sided, and then by whether

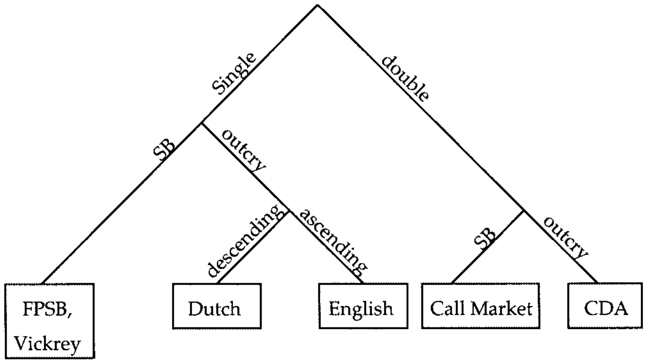

FIG. 1. A classification of classic auction types.

they are sealed-bid or open-outcry. The ascending/descending distinction differentiates the English and Dutch outcry auctions.

Explicit tree representations, however, introduce an artificial ordering on design decisions. In addition, any given tree obscures common features on different branches. Rather than use a hierarchical taxonomy, we focus on the features that define the commonalities and differences among various auctions.

2.1. *Three Core Activities*

The first step in organizing the design space is to recognize three core activities common to auctions. All auctions must perform the first two activities, and most perform the third as well. In the course of an auction, activities of these types may be interleaved and iterated any number of times, depending upon the auction rules.

- **Receive bids**: *Bids* are the messages sent by agents to indicate their willingness to participate in exchanges. On receiving a bid, the auction verifies that it satisfies the auction rules, and if so, *admits* it into the active set of bids.

- **Clear**: The central purpose of an auction is to *clear* the market, determining resource exchanges and corresponding payments between buyers and sellers.

- **Reveal intermediate information**: Auctions commonly supply agents with some form of intermediate status information, typically in the form of hypothetical results were the auction to clear at that moment. We refer to these status reports generically as *quotes*.

This perspective leads naturally to a description of auction features along three axes: bidding rules, clearing policy, and information revelation (quote) policy. Section 3 elaborates on these three axes.

2.2. *The Semantics of Bids*

The canonical bid in an English open-outcry auction represents an offer by the bidder to purchase a unit item at the stated price. Presumably, the bidder will be happier to get the item at an even lower price. Above the stated price, the bidder has not expressed a willingness to buy anything.

To generalize the concepts to multidimensional mechanisms, we consider an auction that allocates the set of resources $\mathcal{J}$. Let $\mathbf{z}$ denote a vector of resource quantities, one for each $j \in \mathcal{J}$. Let $\mathbf{z}_i$ be agent i's *net allocation* vector, with elements $z_{i,j}$ for each $j \in \mathcal{J}$. Element $z_{i,j}$ is the quantity of resource j that agent i is buying/selling in the auction—when $z_{i,j}$ is positive the agent is buying j, and when $z_{i,j}$ is negative, the agent is selling j. The domain of net allocation vectors (assumed the same for every agent) is $Z_j \subseteq \Re$. When Z_j is the set of integers, we describe the resource as *discrete*. When Z_j is an interval of $\Re$, we refer to the resource as *continuous*. Allocation vectors may mix discrete and continuous resources. The domain of $\mathbf{z}$ is the cross product of the individual resource domains, $Z = \prod_{j \in \mathcal{J}} Z_j$.

Let p_j be the price of j expressed in monetary units, and let $\mathbf{p}$ be the vector of all prices. The *net payment* by agent i, π^i, is simply the product of the prices and the quantities in i's net allocation,

$$\pi^i = \mathbf{p} \cdot \mathbf{z}_i. \tag{1}$$

Notice that, whereas prices are nonnegative, net payments can be positive or negative. A negative payment indicates that money is flowing *to* the agent.

Nonlinear pricing relaxes (1). A complete specification of nonlinear prices associates a payment, $\pi_{\mathbf{z}}$, for every allocation vector, $\mathbf{z}$, defining a *payment lattice*, $\boldsymbol{\pi}$. We require that payments be nondecreasing in $\mathbf{z}$ and that the payment associated with the null allocation be zero.

A bid is a message that states an agent's willingness to exchange money —expressed in terms of prices or net payments—for one or more net allocations. The content of the bid (i.e., the *offer*) is a correspondence between net allocations and either prices or payments. When agent i's offer is expressed in prices, we denote it by w_i. When it is in terms of payments, we denote it by ξ_i. Note that the former is a special case of the latter—a vector of quantities specifies a net allocation and the product of quantities and prices determines a payment.

The bid is a reflection of the agent's demand for the resource. In the particular case of competitive analysis, an agent's *Walrasian demand correspondence* assigns the set of utility-maximizing allocations to every price vector $\mathbf{p}$ (Mas-Colell *et al.*, 1995, p. 23). In tatonnement-like protocols, such as the WALRAS algorithm (Cheng and Wellman, 1998; Wellman, 1993), a competitive agent's bid in a (one-dimensional) auction is its demand correspondence assuming the prices of the remaining resources are fixed.

2.2.1. *Bids in Price Space*

Let w_i be i's bid, expressed as a correspondence between prices and quantities. The term $w_i(\mathbf{p})$ represents the set of net allocations that i expresses a willingness to accept at price vector $\mathbf{p}$. Its inverse, $w_i^{-1}(\mathbf{z})$, denotes the set of price vectors for which i would accept the net allocation $\mathbf{z}$.

In order to define bid relationships, we require some notation for comparison between sets. Let Ω be a set with a partial preorder, $\geq$, and let Ω' and Ω'' be subsets of Ω. We say that $\Omega' \gg \Omega''$ iff: (i) for all $\boldsymbol{\omega}' \in \Omega'$, there is an element, $\boldsymbol{\omega}'' \in \Omega''$ such that $\boldsymbol{\omega}' \geq \boldsymbol{\omega}''$, and (ii) for all $\boldsymbol{\omega}'' \in \Omega''$, there is an element, $\boldsymbol{\omega}' \in \Omega'$ such that $\boldsymbol{\omega}' \geq \boldsymbol{\omega}''$. In other words, every element of Ω' has a lesser element in Ω'', and every element of Ω'' has a greater element in Ω'.

A bid is *monotone in prices* if its offer correspondence is nonincreasing in the sense of $\gg$.

DEFINITION 2.1. w_i is monotone in prices iff $\hat{\mathbf{p}} > \mathbf{p}$ implies $w_i(\mathbf{p}) \gg w_i(\hat{\mathbf{p}})$.

Figure 2 illustrates monotone and nonmonotone continuous bids in a single price. The next example considers monotone discrete offers in two dimensions.

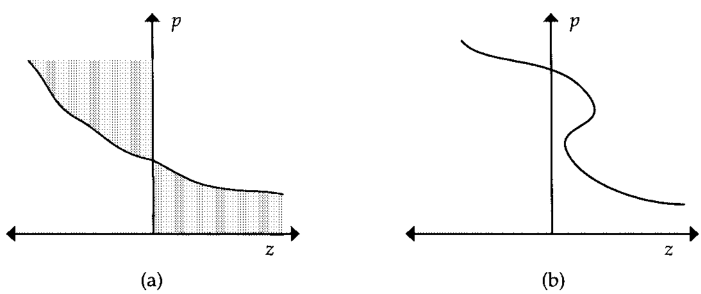

FIG. 2. Examples of continuous bids that are (a) monotone and (b) nonmonotone in prices. The shaded area in (a) indicates the correspondence when the bid is divisible.

EXAMPLE 2.1. Consider a mechanism for two resources, with price vector denoted (p_A, p_B) and net allocations denoted $(z_{i,A}, z_{i,B})$. Suppose that an agent's offer states that $w_i((1,2)) = \{(4,3),(5,2)\}$. Values of the correspondence at the price vector (2, 2) that are consistent with monotonicity include $w_i((2,2)) = \{(4,2)\}$ or $w_i((2,2)) = \{(3,3),(5,1)\}$.

A bid is *divisible* if whenever some allocation is acceptable at a price, all fractions of that allocation are also acceptable. To express this formally, let $\Lambda(\mathbf{z})$ be the set of vectors in Z between the zero vector and $\mathbf{z}$. That is,

$$\Lambda(\mathbf{z}) = Z \cap \{\alpha \mathbf{z} \mid \alpha \in [0,1]\}.$$

DEFINITION 2.2. The bid w_i is divisible if, for all $\mathbf{p}$,

$$\mathbf{z} \in w_i(\mathbf{p}) \quad \Rightarrow \quad \Lambda(\mathbf{z}) \subseteq w_i(\mathbf{p}).$$

Examples of monotonicity and divisibility in discrete bids are shown in Fig. 3. The offer in Fig. 3b is indivisible: although $w_i(p^*) = \{2,3,4\}$, it does not include $z = 1$. The continuous bid of Fig. 2a is divisible if the shaded area is included in the correspondence; it is indivisible otherwise.

If a bid is monotone and divisible, then the offer correspondence can be specified conveniently in terms of its boundary, rather than enumerating the entire set of acceptable net allocations for each price.

2.2.2. *Bids in Payment Space*

We now consider bids expressed as a correspondence between payments and net allocation vectors, focusing on the monotonicity property. A bid is *monotone in payments* if its offer correspondence requires greater allocations as payments increase.

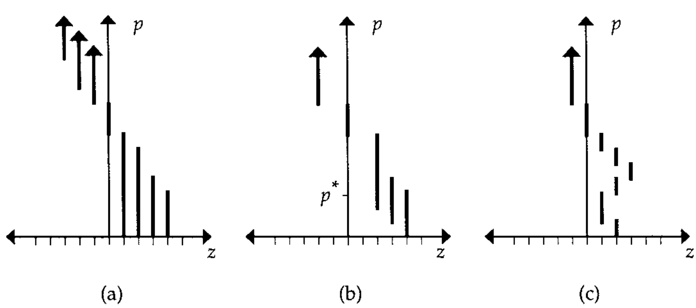

FIG. 3. Examples of discrete bids that are (a) monotone and (b) monotone but not divisible, (c) not monotone.

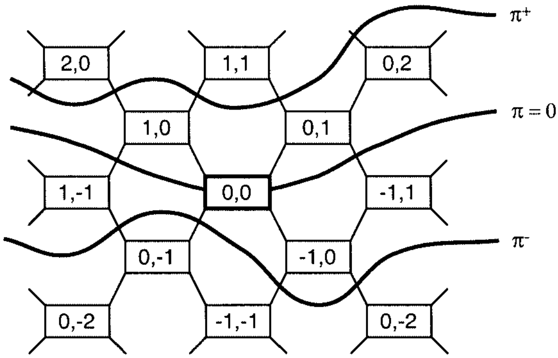

FIG. 4. Iso-payment curves on a lattice.

DEFINITION 2.3. The offer ξ_i is monotone in payments, if, for all $\boldsymbol{\pi}$, $\mathbf{z}$, $\hat{\boldsymbol{\pi}} > \boldsymbol{\pi}$, and $\hat{\mathbf{z}} > \mathbf{z}$,

$$\mathbf{z} \in \xi_i(\hat{\boldsymbol{\pi}}) \quad \Rightarrow \quad \mathbf{z} \in \xi_i(\boldsymbol{\pi})$$

$$\mathbf{z} \in \xi_i(\boldsymbol{\pi}) \quad \Rightarrow \quad \hat{\mathbf{z}} \in \xi_i(\boldsymbol{\pi}).$$

In other words, if $\mathbf{z}$ is acceptable at some payment, then the agent is willing to take $\mathbf{z}$ at a lesser payment, or a greater net allocation at the same payment. Figure 4 illustrates *iso-payment curves* (i.e., curves that connect vectors of equal valuations) on a two-dimensional lattice. The curve labeled π^+ represents a positive payment by the agent. Net allocations below this curve, such as $(0, 1)$, indicate that the agent is not willing to pay π^+ for that change in its allocation. Net allocations above this curve are acceptable to the agent for the payment π^+. The curve labeled $\pi = 0$ represents a net payment of zero, and must intersect the zero allocation vector.

Figure 5 illustrates iso-payment curves on a two-dimensional lattice for a nonmonotone offer. The agent is willing to exchange $(1, -1)$ for a payment π^+, but is not offering to take $(1, 0)$ at the same payment.

Note that monotonicity in prices does not imply monotonicity in payments, nor vice versa. The following examples with discrete offers illustrate the point.

EXAMPLE 2.2. Consider a one-dimensional offer for a discrete good. Suppose i offers to buy (up to) $z_{i,j}$ at a price p_j, and (up to) $\hat{z}_{i,j}$ at $\hat{p}_j$, where $\hat{p}_j < p_j$. The payments associated with $z_{i,j}$ and $\hat{z}_{i,j}$ are $\pi_{z_{i,j}} = p_j z_{i,j}$ and $\hat{\pi}_{z_{i,j}} = \hat{p}_j \hat{z}_{i,j}$, respectively. Monotonicity in prices requires $z_{i,j} \leq \hat{z}_{i,j}$, but it does not impose monotonicity in the payments. For instance, if

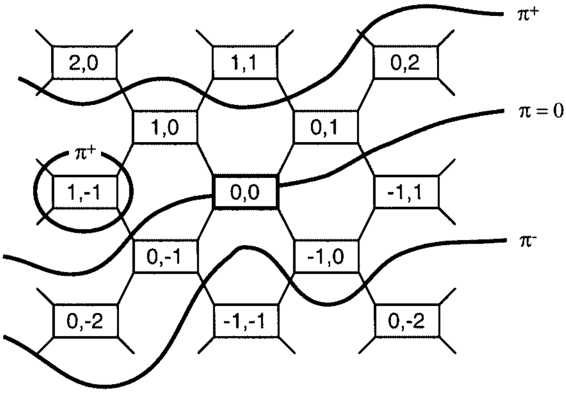

FIG. 5. Nonmonotone iso-payment curves on a lattice.

$z_{i,j} = 10$, $p_j = 4$, and $\hat{p}_j = 2$, any $\hat{z}_{i,j} \geq 20$ is consistent with both monotonicity in prices and monotonicity in payments. However, for $\hat{z}_{i,j}$ between 10 and 20, $\pi_{z_{i,j}} > \hat{\pi}_{z_{i,j}}$, and therefore $\hat{z}_{i,j} \notin \xi_i(\pi_{z_{i,j}})$.

EXAMPLE 2.3. Monotonicity in payments requires $\hat{\pi}_{\hat{z}_{i,j}} \geq \pi_{z_{i,j}}$ when $\hat{z}_{i,j} \geq z_{i,j}$, but does not impose monotonicity in prices. Suppose at $z_{i,j} = 10$, $\pi_{z_{i,j}} = 40$, and at $\hat{z}_{i,j} = 11$, $\hat{\pi}_{\hat{z}_{i,j}} = 55$. The former case maps to $p_j = 4$, while the latter maps to $\hat{p}_j = 5$. However, $w_i(p_j) \not\succeq w_i(\hat{p}_j)$.

3. THE AUCTION PARAMETER SPACE

In this section, we present a parametrization of the auction design space that is broad enough to encompass most of the classic auctions, common commercial and online auctions, and many others. We have attempted to define the parameters in such a way as to make them orthogonal. One-dimensional versions of most of these parameters have been implemented in the AuctionBot system, providing users great flexibility in choosing the rules of the auctions they create. We are currently in the process of extending the AuctionBot to support various forms of multidimensional auctions.

Earlier, sparser versions of this parametrization appear in previous reports (Mullen and Wellman, 1996; Wurman *et al.*, 1998b). Other researchers have also attempted to organize the space of auction designs. Engelbrecht-Wiggans (1980) parametrizes a smaller set of auctions as part of a broad classification of auction research. Friedman (1993) presents a taxonomic structure of the design space with particular emphasis on variations of the CDA. The FM96.5 (FishMarket) testbed (Rodríguez-Aguilar *et al.*, 1998) is based on a detailed parametrization of the space of Dutch auctions. IBM developed an Internet auction server that implemented a subset of the parameters described below in the context of single-unit auctions (Kumar and Friedman, 1998). None of these previous works, including our own, considered multidimensional auctions or bids in the form of arbitrary correspondences.

Our presentation is organized along the three axes introduced in Section 2.1: bidding rules, clearing policy, and information revelation policy.

3.1. *Bidding Rules*

Bidding rules determine under what conditions bids may be introduced, modified, or withdrawn, as a function of agent identity, current bid status, or even the entire auction history. If an incoming bid satisfies these rules, then the auction *admits* it into the set of current bids. If a bid fails to satisfy the admission criteria, the auction notifies the agent that the bid was rejected.

Because bids can express arbitrary correspondences, we can assume (without loss of generality) that each agent has at most one active bid in an auction at any time. An agent changes its bid by submitting a new one, if allowed by the auction rules.

3.1.1. *Expressiveness*

An auction mechanism dictates a *language* for bids, defining their syntax as well as expressive power. In this discussion we are concerned with semantics, and thus limit attention to expressive power. The bid language dictates whether offers are in terms of prices or payments, and the class of correspondences that may be expressed.

For example, classes of correspondences supported by bid languages in the Michigan Internet AuctionBot include the following:

- **Price-quantity schedules**. A bid schedule is a stepwise specification of offers to buy or sell various quantities at discrete price points. The price-quantity expression has two subclasses, corresponding to restrictive special cases:

  —**Single units**. An agent can express a single-unit offer concisely as a price and a sign indicating whether the offer is to buy or sell.

  —**Single price points**. This option restricts offers to a fixed number of units (buy or sell) at a single, per-unit price.

- **Continuous**. Using analytical expressions, a bid may specify offers that are continuous functions of prices.

- **Combinatorial**. When the auction is multidimensional and mediates the allocation of discrete goods, agents make offers on bundles of resources. Bids are correspondences between bundles and payments.

General properties of correspondences, such as those mentioned in Section 2.2, can be used to further restrict bid expressions. For example, an auction might require that bids be monotone or divisible. When they are divisible (whether required or not), then a price-quantity schedule or continuous offer function would typically be interpreted as a boundary on the set of acceptable allocations in the offer correspondence.

Restrictions on bid expressions may reflect domain constraints, or auction policies adopted for reasons of computation, communication, or incentive engineering. For example, single price points may be considered easier to specify than complete schedules. As another example, requiring that offers be divisible may simplify clearing calculations (discussed in Section 4). In the combinatorial context, Rothkopf *et al*. (1998) present several cases where restricting the expressive power of bids allows polynomial-time computation of optimal allocations in multidimensional auctions for discrete resources.

3.1.2. *Buyers and Sellers*

Typically, we classify auctions by whether they have one buyer or many buyers, and one seller or many sellers. The three combinations of interest are {one buyer:many sellers}, {many buyers:one seller}, and {many buyers: many sellers}. A restriction to "one" essentially means that the auction is one-sided, and the sole buyer or seller must be designated.

For example, a one-dimensional auction with a single buyer, designated h, imposes the restrictions

$$\forall i \neq h \qquad \forall p_j, z \in w_i(p_j) \qquad z \leq 0$$
$$\forall p_j, z \in w_h(p_j) \qquad z \geq 0.$$

Similarly, if h is the designated sole seller,

$$\forall i \neq h \qquad \forall p_j, z \in w_i(p_j) \qquad z \geq 0$$
$$\forall p_j, z \in w_h(p_j) \qquad z \leq 0.$$

The restrictions can be extended in a straightforward manner to bids expressed in payments. Note that in the {many buyers:many sellers} case, agents' bids can express offers to either buy or sell, or both.

In the multidimensional case, each agent's bid may be permitted to express willingness to buy or sell, or both, for each resource type. An even more general formulation of this rule allows the auction to restrict each agent's bids to certain ranges of the offer correspondence space.

3.1.3. *Dominance*

Bid dominance rules restrict the relationship of an agent's new offer to the bid, if any, it replaces. Both increasing and decreasing forms of dominance may apply, and, when bids are monotone, an auction can impose different restrictions on the buy and sell sides of the bid.

Let w_i and $\hat{w}_i$ be two bids.

DEFINITION 3.1. $\hat{w}_i$ is *superior* to w_i in the range $[\underline{\mathbf{p}}, \overline{\mathbf{p}}]$ iff, for all $\mathbf{p} \in [\underline{\mathbf{p}}, \overline{\mathbf{p}}]$, $\hat{w}_i(\mathbf{p}) \gg w_i(\mathbf{p})$.

DEFINITION 3.2. $\hat{w}_i$ is *inferior* to w_i in the range $[\underline{\mathbf{p}}, \overline{\mathbf{p}}]$ iff w_i is superior to $\hat{w}_i$ in that range.

Figure 6 illustrates a bid and three alternative bids. Bid w' is superior to w over all prices, w'' is inferior over all prices, and $\tilde{w}$ is inferior in the prices that correspond to negative quantities, and superior in positive quantities.

The *ascending rule* requires that new bids be superior to old bids, while the *descending rule* requires new bids be inferior. Intuitively, the combination that requires that the buy side of a bid increases and the sell-side of the bid decreases is the most natural, since it represents a strengthening of the offer on both sides. However, the rule that requires a seller to *increase* its sell bid has found use as part of a protocol for decentralized task allocation (Walsh and Wellman, 1998). It is an example of how an

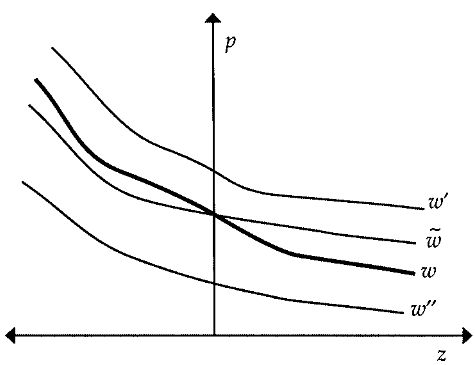

FIG. 6. Examples of continuous bids that are superior, inferior, and mixed with respect to the original bid.

expanded view of the auction design space can lead to new, potentially interesting protocols.

Note that the increasing restriction, when applied to the sell side, allows an agent to effectively withdraw its bids by offering to sell at infinity. Similarly, when the decreasing rule is applied to the buy side, the agent can effectively withdraw by offering to buy at a price of zero.

For bids expressed in payment space, $\hat{\xi}$ is superior to ξ if it expresses maximal payments at least as great on all net allocation vectors. For monotone bids, this is the case iff, for all π, $\hat{\xi}_i(\pi) \supseteq \xi_i(\pi)$. That is, the set of net allocations acceptable at the payment π expressed by $\hat{\xi}$ must be a superset of those expressed by ξ.

It is useful to compare the effects of increasing a bid in price space on the representation of that bid in payment space. Consider two points on a price-quantity schedule: (z^1, p^1) and (z^2, p^2), where $z^1 < 0 < z^2$ and $p^1 > p^2$. A new bid $\hat{w}$, which is superior (in prices) to w, increases the price at which the agent is willing to exchange the respective quantities. Projected into payment space, $\hat{\pi}_{z^1} < \pi_{z^1}$ (because z^1 is negative), and $\hat{\pi}_{z^2} > \pi_{z^2}$. Thus, the ascending price rule causes the bid to *increase* on the buy side and *decrease* on the sell side in payment space.

Thus, because of the sign change, the direction of influence of the ascending rule in payments is the same as the mixed rule in prices. Figure 7 illustrates the four offers in Fig. 6 mapped into payment space. $\tilde{\xi}$, which had mixed dominance in prices, is superior to ξ in payments.

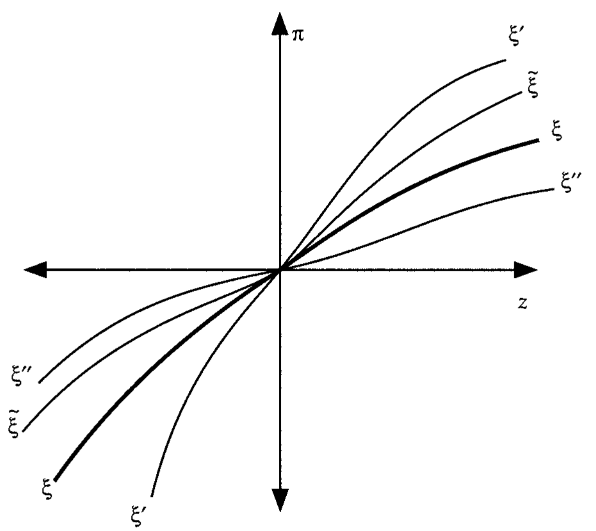

FIG. 7. The four price-space bids from Fig. 6 mapped into payment space.

3.1.4. *Beat-the-Quote*

The dominance rules of the previous section require that new bids bear some relation to previous bids by the same bidder. Many auctions require that a new bid satisfy some condition defined in terms of bids by other bidders. For example, in an English outcry auction, a new bid must beat the highest so far (perhaps by a specified increment). We capture this concept by defining analogous dominance rules with respect to *price quotes*, summary information about the bid state revealed by the auction (discussed in detail in Section 3.3.1). Typically, the purpose of such rules is to ensure a progression of prices, thus directing the process to a steady state, if not an actual equilibrium.

Typically, the auction's quote provides evidence regarding potential outcomes of the auction, in the form of the agent's tentative allocation. Let $\chi_i(q, w)$ denote the allocation that price quote q entails agent i would receive had it sent the bid w to the auction.[5]

Again, let w_i be the agent's current bid, and let $\hat{w}_i$ be its new bid. The one-dimensional beat-the-quote rule requires that

$$\begin{aligned} \text{if } \chi_i(q, w_i) > 0, \qquad & \chi_i(q, \hat{w}_i) \geq \chi_i(q, w_i) \\ \text{else if } \chi_i(q, w_i) < 0, \qquad & \chi_i(q, \hat{w}_i) \leq \chi_i(q, w_i). \end{aligned}$$

In words, an agent's new bid must keep the same sign of, and must not decrease the magnitude of, its tentative allocation. The rule for payment space is the same, with ξ substituted for w. Using strict inequalities, we can formulate strict versions of either beat-the-quote rule. Note that it would not have been sufficient to define this rule in terms of simple price/payment increases (or decreases), since bids in general refer to various quantities at various prices/payments.

EXAMPLE 3.1. Consider a one-dimensional auction for discrete resources in which an agent has submitted an indivisible bid. Suppose the agent has a current offer to sell two units if the price is between \$5 and \$10, or to sell four units if the price is \$10 or greater. A price quote of \$8 implies that the agent is winning two units. A new offer to sell two units if the price is between \$5 and \$7, or four units if the price is \$7 or greater, would increase the agent's tentative winnings to four units, and thus satisfies the beat-the-quote rule. A new offer to sell two units if the price is between \$3 and \$10, and four units if the price is \$10 or greater, would

[5] This interpretation is typical, but not required. That is, a beat-the-quote rule is well defined for any coherent interpretation of $\chi_i(q, w)$ as long as this function itself is well defined.

satisfy the nonstrict version of the beat-the-quote rule, but not the strict version.

Notice that the bid dominance and strict beat-the-quote rules are complementary. Both potential new bids in Example 3.1 dominate the original bid, but only the first strictly beats the quote. The next example demonstrates that a new bid can strictly beat the quote without dominating the previous bid.

EXAMPLE 3.2. Consider an agent with a divisible offer to buy two units at \$2, or one at \$10. Suppose the price quote is \$4. A new bid, in which the two-unit offer is at \$5 and the one-unit offer at \$7, would beat the quote, but would not satisfy dominance because the offer decreased from one to zero units in the range [\$7, \$10].

We can generalize this to the multidimensional case by requiring that it hold with respect to each resource. This requirement may be too strong, however, as evidenced by the following example.

EXAMPLE 3.3. Consider a multidimensional auction for two single-unit resources. There are four possible net allocations: $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$. Suppose an agent has bid \$2 on $(1,0)$, \$4 on $(0,1)$, and \$7 on $(1,1)$. The auction announces the nonlinear price quote $\{\pi_{(1,0)} = 3,\ \pi_{(0,1)} = 3,\ \pi_{(1,1)} = 8\}$. This quote implies that the agent is currently winning $(0,1)$. It is quite possible that the agent would desire to raise its bid on $(1,0)$ next. However, the version of the beat-the-quote rule described above would not permit this switch because the quantity of the second resource decreases.

An alternative multidimensional generalization would require only that the beat-the-quote condition hold with respect to *some* resource. This may seem rather weak but, in conjunction with a dominance rule, it is often exactly what is desired. For example, several of the combinatorial auctions mentioned in Section 4.4 essentially apply this combination of bid restrictions.

The most common variant of the beat-the-quote rule requires that the new bid increase the price by some increment δ above (or below) the quoted prices. For example, in the English auction, when the agent is not winning the good, its new offer must not only make it the winning bidder, but it must do so at a price at least δ above the current winning bidder. We extend this concept to the more general case in the following manner. New bid $\hat{w}_i$ *beats the quote by* δ iff there exists some w' such that

$$\text{if } \chi_i(q, w_i) > 0,$$

$$\chi_i(q, \hat{w}_i) \geq \chi_i(q, w') \geq \chi_i(q, w_i)$$

$$\hat{w}_i^{-1}(\chi_i(q, w')) \geq w'^{-1}(\chi_i(q, w')) + \delta$$

else if $\chi_i(q, w_i) < 0$,

$$\chi_i(q, \hat{w}_i) \leq \chi_i(q, w') \leq \chi_i(q, w_i)$$

$$\hat{w}_i^{-1}(\chi_i(q, w')) \leq w'^{-1}(\chi_i(q, w')) - \delta.$$

Such a rule could similarly be formulated for payment space.

The beat-the-quote-by-δ rule is used to speed the progression of prices. There is a risk that the rule will result in a loss of efficiency when an agent values the object more than the winner, but not by enough to continue bidding. However, in many protocols—such as the multi-item auction of Demange *et al.* (1986)—the efficiency loss is bounded, typically by an amount linear in δ.

3.1.5. *Withdrawal and Expiration Rules*

Auction rules dictate whether bid withdrawals are allowed, and if so when. For example, one possible rule is that withdrawals are permitted only in conjunction with a clearing operation. An expiration is a planned withdrawal, typically specified at bid time, also subject to permissibility by the auction rules.

3.1.6. *Activity Rules*

In many complex domains, agents can benefit from a strategy in which they withhold information while others reveal information. For example, the top two performing agents in the 1990–91 Santa Fe Institute double auction tournament employed a strategy of waiting in the background while others bid. When the market seemed about to converge, the agents would step in and "steal the deal" (Rust *et al.*, 1994). Clearly, if everyone employed this strategy, the market would fail. Activity rules are designed to counteract such behavior.

DeMartini *et al.* (1998) present a formal definition of one feasible activity rule.[6] In each round, an agent is eligible to place bids on as many items as it had (provisionally) winning bids, plus the number of bids it placed in the previous round. For example, if the agent was provisionally winning items A and B last round, and it chose to enhance its bid with offers on items C and D, then it is eligible to bid on up to four items.

Systematic parametrization of the space of activity rules would be quite useful, and is a topic for future work.

[6] The rule described was employed in the simultaneous ascending auction used by the FCC and is part of the proposed RAD mechanism.

3.2. *Clearing Policy*

As noted in Section 2.1, an auction *clears* when it commands an allocation based on the bids it has received. We formalize the auction's clearing policy in terms of a *matching function*, γ, which maps the set of current bids to net allocations and payments for the agents involved. By calling it a matching function, we underscore that the array of net allocations corresponds to an exchange of resources (and money) among the agents. This section details the parameters that dictate how and when this exchange is determined.

3.2.1. *Matching Function*

We can view the matching function as determining the exchanges in two steps: determine first which agents will trade and, second, the exact terms of each exchange. Define the *surplus* of an exchange to be the difference in maximal (resp. minimal) payments the agents are willing to pay (receive) for the given net allocation, as expressed by their bids. For example, if one agent offers to buy a unit of the good for p^1, and another offers to sell at p^2, the surplus is $p^1 - p^2$. If its surplus is positive, the exchange is considered mutually beneficial. We apply the term *clear* to the operation of determining allocations because in general, after the exchanges are executed, no mutually beneficial trades exist among the offers represented by remaining bids. An allocation is *locally efficient* if it maximizes the total surplus as represented by the bids.

EXAMPLE 3.4. Consider two candidate allocations for the single-unit bids listed in Table I. In the first, agents 1 and 4 trade. This trade allocates the goods to the agents with the highest stated values—agent 4 buys one and agent 3 keeps one. In the second potential allocation, agent 4 trades with agent 3, and agent 2 trades with agent 1. The agents who end up with the goods have the first and third highest valuations. This allocation is inefficient because agents 2 and 3 could both be made better off by a further trade between them. Although both sets of exchanges clear the

TABLE I
An Example with Four Bids

Agent	Offer
Agent 1	sell 1 unit at \$1
Agent 2	buy 1 unit at \$2
Agent 3	sell 1 unit at \$3
Agent 4	buy 1 unit at \$4

market (i.e., there are no trades available among the remaining bids), only the former is locally efficient.

In general, there may be many allocations that satisfy local efficiency but differ in their allocation of money.[7] For example, the exchange that results in the locally efficient solution for Example 3.4, namely, that agent 1 sells a unit to agent 4, would be mutually beneficial if agent 4 paid agent 1 any price between $1 and $4 for the object. At the lower end of the range, agent 4 captures all $3 of the surplus. At the higher end of the acceptable price range, agent 1 captures the surplus.

The selection of matching functions depends on the nature of the resource, the dimensionality of the auction, the expected format of the agents' utility functions, and the designer's goals. A variety of matching functions are discussed in Section 4.

3.2.2. *Clear Timing*

This important parameter of the clearing policy determines *when* clears should occur. Some common forms of timing policy include:

Scheduled. Clears occur at a specified set of nominal times. The specification may take the form of an explicit enumeration, or some implicit description, for example, the frequency of a periodic clear.

Random. Clears are determined according to some random distribution. Memoryless distributions, such as the Poisson, are attractive candidates because they deter agents from applying complex time-dependent strategies. The relevant parameters of the distribution are typically made public, so the agents participate with some expectation of the timing of events.

Bidder activity. Clears occur whenever a new bid is admitted. This is the timing policy of *continual* auctions such as the CDA. A variation of this rule allows for *synchronized* auctions—rather than clearing when a bid is admitted, the auction can clear when a new bid has been received from each participant, or when a fixed number of bids has been received.

Bidder inactivity. The auction clears when no bid has been admitted for a specified period.

[7] There may also be multiple locally efficient solutions that differ in their distribution of resources, for instance, if there are tie bids.

Auctions can also combine these schedules in various ways. For example, some online versions of the English auction clear at a time determined by the latest of a scheduled time and a period of inactivity.

3.2.3. *Closing Conditions*

The closing conditions are logical tests that determine whether a clear should be the final clear. Auctions can close at a scheduled time, at a random time, after a period of inactivity, or when the bids of designated agents (e.g., the seller, in a single-seller auction) are matched. In some cases, we desire that auctions close when an external signal is received (e.g., indicating that some more global quiescence property has been achieved; Wellman and Walsh, 2000).

3.2.4. *Tie Breaking*

A *tie* occurs whenever two agents express a willingness to take the same net allocation at the same price (or payment). The manner in which ties are resolved can also influence the outcome of an auction. Three common methods are to break ties arbitrarily, in favor of the earlier bids, or in favor of bids for larger quantities.

3.2.5. *Auctioneer Fees*

The policies discussed thus far determine payments between the buyer and seller. In commercial auctions, it is common for the auctioneer to collect some fees, and these payments must be considered by the agents. A payment may be required of the buyer, the seller, or both. Common types of transaction fees include:

Entrance fee. A fixed fee for the agent's first bid. This is a generalization of a listing fee.

Bid fee. A fixed fee paid with every bid. Bid fees can provide a disincentive to price manipulation (McCabe and Smith, 1993).

Ad valorem. A percentage of the exchange price.

Nonlinear. A nonlinear function of the exchange price, such as those often used to provide quantity discounts.

These are just some of the possible fee structures. When an outside subsidy is present, these fees can flow the opposite direction—from auctioneer to agents. Anderson et al. (1999) explore the incentive properties of *k*-double auctions (Satterthwaite and Williams, 1999) with various fee structures.

3.3. *Information Revelation Policy*

We formalize the auction's information revelation policy in terms of a *quote function*, ρ, which maps the set of current bids to a message we call the *price quote*, which represents some summary of the current bid state.[8] The parameters discussed below control the timing and content of price quotes.

3.3.1. *Price Quotes*

The feature common to price quotes in many classic auctions is their information content. A price quote informs an agent of the range of offers that would have been in the exchange set had the auction cleared at the time the quote was issued. This definition is given in past hypothetical tense because a price quote is necessarily relative to the bidding state at quote time—if the time necessary to transmit the price quote message varies between agents, the price quote may be stale before it is received.

In many cases, the quote function ρ can be viewed as an impotent version of the matching function γ—it calculates hypothetical exchanges and then announces the prices (or payments) without actually clearing. For example, the English auction uses the first-price rule to both generate price quotes and determine the exchange when the auction clears. We say the quote is *separating* if all agents can correctly infer their tentative allocations. Using the notation employed in Section 3.1.4, a separating quote provides perfect information $\chi_i(q, w)$ about the allocation i gets with current bid w.

However, it is not necessary to use γ as a basis for ρ. Rassenti et al. (1982) and DeMartini et al. (1998) both present combinatorial auctions that use relaxed linear programming techniques for ρ and integer programming for γ. In both cases, the information provided by the quote q is not always enough for an agent to correctly infer its allocation. If q is *noisy*, then $\chi_i(q, w)$ may correspond to a distribution over potential allocations.

When the same quote is reported to every agent, it is *anonymous*. When quotes are customized for each agent, we say they are *discriminatory*, and denote them q^i.

In one-dimensional auctions with continuous, strictly monotone bids, the hypothetical clearing price, necessarily, must be the price, p^*, that balances aggregate supply and demand. In one-dimensional auctions for discrete resources with divisible bids, there is typically a range of clearing prices, specified by the endpoints $\bar{p}$ and $\underline{p}$. The *bid quote*, $\underline{p}$, is the price an

[8] Typically, the quote is dependent on only the current bids, but this assumption could be relaxed and the entire history of bids could be used as input to the quote function.

agent would have to bid under in order to place a winning sell bid. The *ask quote*, $\bar{p}$, is the price an agent would have to bid over to place a winning buy bid.

In the CDA, standing buy and sell offers never overlap; hence the bid and ask prices reflect the spread between the highest buyer and the lowest seller. More generally, the bid and ask prices have meaning even if the buy and sell offers do overlap. Consider a set of L single-unit bids, of which M are sell offers and the remaining $N = L - M$ are buy offers. Sort all bids by price, and count down the list of bids. The Mth unit determines $\bar{p}$, the upper bound of the separating price range, and the $(M + 1)$st unit determines $\underline{p}$, the lower bound (Wurman *et al.*, 1998a).[9] In fact, the Vickrey auction (Vickrey, 1961) is an examination of the special case where $M = 1$: $\bar{p}$ corresponds to the first price, and $\underline{p}$ corresponds to the second price.

EXAMPLE 3.5. Consider one agent with an offer to buy one unit at \$10, and another with an offer to sell one unit at \$5. The buy and sell offers overlap, so if the auction cleared at this moment, the two offers would transact. A bid–ask quote, in this case, would report that a new buyer would need to outbid \$10 (displacing the current buyer), and a new seller underbid \$5 (displacing the current seller), in order to be part of the tentative allocation.

Under some conditions, if an agent's bid is equal to the bid or ask quote, it cannot tell whether it is in the exchange set. Thus, to produce a separating quote, the auction may need to augment the bid–ask quote with information directly telling the agent its tentative allocation given the current bids.

When multiunit indivisible bids are allowed, separating prices may not exist. That is, there may not be a single per-unit price above which an agent's buy offer is accepted and below which it is rejected. For example, two of the bids shown in Table II are indivisible. If the per-unit price is set at or below \$4, then the resources are overdemanded. If the price is set above \$4, the resources are underdemanded. For instance, consider the case where the price is set to \$4.5 and one unit is sold to agent 3. What is particularly unappealing about this solution is that it is Pareto dominated by the allocation where agent 2 buys the remaining two units at a price less than \$2. Moreover, the socially efficient solution is to sell all three units to agent 4. Attempting to determine an exchange set based on a single price fails to exploit all profitable trades.

In order to present the agents with the information necessary to determine their hypothetical allocations, the auction needs to discriminate

[9] The ask quote is undefined if there are no sell offers, and the bid quote is undefined if there are no buy offers.

TABLE II
An Example with Indivisible Bids without an Efficient Clearing Price

Agent	Offer	Divisible?
Agent 1	sell 3 units at $1	Yes
Agent 2	buy 2 units at $2	No
Agent 3	buy 1 unit at $5	Yes
Agent 4	buy 3 units at $4	No

based on quantity. Nonlinear pricing allows an auctioneer faced with the example in Table II to quote, for example, $3.5 for three units, $4.5 for two units, and $5.5 for one unit. These prices support the efficient allocation and clearly indicate to the agents whether they are in the tentative exchange set.

Elsewhere (Wurman and Wellman, 1999), we have shown that a lattice of separating payments always exists in one-sided combinatorial auctions. The method of constructing these payments is the basis of the A1BA mechanism (Wurman, 1997, Chap. 5), discussed briefly in Section 4.4.

Before leaving the topic of price quotes, we should mention price clocks, which are used in variations of the Dutch auction. A specification of a price clock must include start and end prices, and a (usually linear) price adjustment schedule. Clocks can be used to generate the bid quote, the ask quote, or both. Although the implementation of online Dutch auctions has been investigated (Rodríguez *et al.*, 1997), considerable infrastructure is required to ensure that all agents have equal access to the auction over the distributed and asynchronous network. We argue elsewhere (Wellman and Wurman, 1998b) that little is gained by attempting to reconstitute real-time auctions online; our effort is better spent developing auctions that take advantage of the asynchrony and flexibility of the online environment.

3.3.2. *Quote Timing*

Like clear events, price quotes can vary in number and frequency. Some of the most significant choices are:

No price quotes. Auctions that reveal no information are traditionally called sealed-bid auctions.

Scheduled. Quotes are generated according to a specified nominal schedule.

Random. Price quotes are generated according to some stochastic process.

Bidder activity. Price quotes are generated with each new bid admitted.

Bidder inactivity. Price quotes are generated when no bid has been admitted for a specified period.

An auction may generate many price quotes as it proceeds, depending upon the quote schedule and the clearing and closing policies of the auction.

3.3.3. *Order Book*

The term *order book* is commonly used in organized exchanges, like the NYSE, to refer to the current set of active bids. The auctioneer may make some or all of the information in the order book public. The most common choices are to keep the book closed, reveal only the current winning bids, or to open the book completely. Online auctions, such as eBay and Onsale, commonly identify tentative winners.

3.3.4. *Transaction History*

Auctions may publicize selected information about past exchanges. Such information may include the prices, quantities, or even the identities of the transacting agents. If an auction has several clears but does not reveal historical prices, then agents get from the auction price information only for exchanges in which they participated. Publicly revealing past transaction prices avoids such an information asymmetry.

4. MATCHING FUNCTIONS

In this section we isolate the matching functions used in a variety of online and experimental auctions.

4.1. *Uniform-Price One-Dimensional Matching Functions*

A clearing policy is *uniform price* if every exchange in a given clear occurs at the same price. Recall that a quote is separating if an agent can correctly infer its tentative allocation. When bids are monotone and divisible, a separating uniform price is guaranteed to exist. In the discrete case, there is typically a range of such prices.

The set of bids in the exchange set can be identified in the following manner. First, for conceptual simplicity, let us treat a multiunit offer as a collection of component single-unit offers. Let m denote the number of unit sell offers at or below $\underline{p}$, and let n denote the number of unit buy offers at or above $\bar{p}$. Let $l = \min(m, n)$. The set of winning buy offers, B_{in}, comprises the l highest unit buy offers. The set of winning sell offers, S_{in}, comprises the l lowest unit sell offers. We justify focusing attention on this method of generating the transaction set because selecting any bids outside of this set would violate local efficiency. A uniform-price auction applying these rules can use any tie-breaking policy and any algorithm for pairing the winning buy and sell bids.

Having selected the winning bids, we now turn our attention to setting the transaction price. The matching function γ^k, used in the k-double auction (Satterthwaite and Williams, 1989; Satterthwaite and Williams, 1993), sets the transaction price according to $p = k\bar{p} + (1-k)\underline{p}$, $k \in [0, 1]$. γ^k covers the full range of transaction prices that are uniform, separating, and support locally efficient allocations.

Notice that the two extreme values of k produce the $(M+1)$st and Mth prices, respectively. Although no uniform-price sealed bid auction is Bayes–Nash incentive compatible for multiunit buyers or sellers (Wurman *et al.*, 1998a), the $(M+1)$st-price, sealed-bid auction is incentive compatible for single-unit buyers, and the Mth-price, sealed-bid auction is incentive compatible for single-unit sellers.

The dual-price mechanism (McAfee, 1992) uses the k-auction pricing but excludes the lowest-priced offer in B_{in} and the highest-priced offer in S_{in}. The dual price mechanism maintains individual rationality, incentive compatibility, and budget balance, but sacrifices local efficiency by eschewing the lowest-surplus trade.

Another uniform-price auction that sacrifices local efficiency in pursuit of other desirable properties is the UPDA (McCabe and Smith, 1993). The "1S" version of the UPDA announces, as its ask quote, the second highest offer in the union of all the non-winning buy bids and the lowest offer in B_{in}. The bid quote is computed in an analogous manner using the sell bids. A new bid must beat the quote in order to be included in the tentative match set. We designate this matching function γ^{1S}. The admission rule results in a book in which the tentative match set may not correspond to the highest buy and lowest sell bids. Instead, membership in B_{in} and S_{in} is dependent on the order bids are received, and the procedure described in this section cannot be used to implement the 1S version of the UPDA.

4.2. *Discriminatory-Price One-Dimensional Matching Functions*

When we relax the constraint that prices be uniform, a wider range of pricing options becomes available. Table III provides an example with which we can compare the matching functions in this section. Here $t_1 < t_2 < t_3 < t_4 < t_5 < t_c$, where t_c is the time the clear occurs.

TABLE III
A Sequence of Five Single-Unit Bids

Time	Agent	Offer
t_1	Agent 1	sell at $5
t_2	Agent 2	buy at $8
t_3	Agent 3	buy at $7
t_4	Agent 4	sell at $6
t_5	Agent 5	buy at $9

Local efficiency demands that agents 1 and 4 be the winning sellers and agents 2 and 5 be the winning buyers. We can determine B_{in} and S_{in} in the manner discussed in the previous section. There are two possible exchange sets that can be formed from this combination of buyers and sellers. Under uniform pricing, all transactions occur at the same price. Thus, agents are indifferent between the two possible exchange combinations. However, as the following discussion shows, when prices are discriminatory, it matters a great deal how agents in the transaction sets are matched.

4.2.1. *Pay Buyer's/Seller's Bid*

There are several ways in which we might consider generalizing the first- and second-price rules. One uses the uniform Mth and $(M + 1)$st-price rules discussed above. An alternative generalization is to require that exchanges occur always at the seller's price or always at the buyer's price. Consider the application of the buyer's-price and seller's-price policies to the example in Table III. Suppose the auction matches the highest buy offer with the lowest sell offer until the exchanges are exhausted. The *pay-seller's-price* policy would result in transactions

agent 1 sells to agent 5 for $5
agent 4 sells to agent 2 for $6.

The *pay-buyer's-price* policy produces

agent 1 sells to agent 5 for $9
agent 4 sells to agent 2 for $8.

We can form a convex combination of the two extremes. Let $\kappa \in [0, 1]$. The price of the transaction between buyer i and seller h is $\kappa w_i + (1 - \kappa)w_h$. Note that, unlike the k-auction function, the κ-price is computed for each pair of agents. This function, denoted γ^{κ}, is used in a study by Hu and Wellman (1998).

4.2.2. *Chronological Pricing*

Chronological pricing uses the bids' submission times to determine the exchange price. Given an exchange pair, *earlier-bid* pricing uses the price of the bid that was placed earlier, whereas *later-bid* pricing uses the price of the later bid. The former, designated γ^{ET}, generalizes the method in which prices are determined in the CDA to permit it to be used regardless of the timing of clears. Later-bid pricing, γ^{LT}, is simply γ^{ET}'s natural complement.

Consider the application of the earlier-bid pricing to the exchanges determined above. Earliest pricing computes the transactions

agent 1 sells to agent 5 for \$5
agent 4 sells to agent 2 for \$8.

If, instead, the auction formed the same matches but set the prices according to the later bids, the following transactions would occur

agent 1 sells to agent 5 for \$9
agent 4 sells to agent 2 for \$6.

4.2.3. *Comparisons*

Table IV summarizes the different transaction prices for seven of the pricing functions described to this point. The simple example problem was constructed to illustrate differences between these seven, and does not

TABLE IV
Transaction Prices for the Seven Matching Functions

	Transactions	
	1 sells to 5 for	4 sells to 2 for
Mth ($\gamma^{k=1}$)	\$8	\$8
(M + 1)st ($\gamma^{k=0}$)	\$7	\$7
Dual price ($\gamma^{k=1}$)	\$8	excluded
Seller's price ($\gamma^{\kappa=0}$)	\$5	\$6
Buyer's price ($\gamma^{\kappa=1}$)	\$9	\$8
Earlier bid (γ^{ET})	\$5	\$8
Later bid (γ^{LT})	\$9	\$6

elicit interesting behavior from γ^{1S}, so that function is excluded from the comparison.

As mentioned, the surplus that an agent captures depends on how the transaction pairs are determined. Table V shows, for the four discriminatory pricing functions, how the price, and therefore agent 4's surplus, depends on who agent 4 trades with.

4.3. *Matching Functions with Indivisible Bids*

Suppose all of the offers are indivisible. The per-unit price at which agent i expresses willingness to exchange z units is given by $w_i^{-1}(z)$. Note that a negative value of z expresses a sell offer, and a quantity that i has not expressed a value for has $w_i^{-1}(z) = 0$ if $z > 0$, and $w_i^{-1}(z) = \infty$ if $z < 0$. The surplus maximization problem can be stated as

$$\max \sum_i \sum_z z w_i^{-1}(z) \delta^{iz} \tag{2}$$

$$\text{s.t.} \quad \sum_i \sum_z z \delta^{iz} \le 0$$

$$\sum_z \delta^{iz} \le 1 \qquad \forall i$$

$$\delta^{iz} \in \{0, 1\},$$

where $\delta^{iz} = 1$ means that i's offer for z units is part of the solution. The first constraint states that we do not allocate more than we have (and can dispose of any extra supply at no cost). The second constraint ensures that no agent wins at more than one quantity level. We designate the matching function that solves this optimization function γ^{knapsack}.

When there is only one sell offer, and it has an associated reserve price of zero, (2) reduces to a 0–1 knapsack problem (Garey and Johnson, 1979). Therefore, (2) is also NP-hard.

TABLE V
In Discriminatory Auctions, Agent 4's Transaction Price Depends on Its Trading Partner

	Transactions	
	4 sells to 2 for	4 sells to 5 for
Seller's price	\$6	\$6
Buyer's price	\$8	\$9
Earlier price	\$8	\$6
Later price	\$6	\$9

TABLE VI
A Set of Bids That Can Lead to Arbitrarily Bad Outcomes when $p_2 > p_3$ and the Greedy Algorithm Is Used

Agent	Offer	Divisible?
Agent 1	sell z units at \$1	Yes
Agent 2	buy 1 unit at p_2	Yes
Agent 3	buy z units at p_3	No

A second problem associated with bid indivisibility is that linear pricing may not admit separating price vectors, as discussed in Section 3.3.1. To provide agents with the information necessary to determine their status in the auction, the auction needs to quote a nonlinear price vector.

In practice, online auction services (e.g., UBid[10]) that allow indivisible bids in a single-seller auction implement a greedy approximation algorithm, γ^{greedy}. Bids are sorted decreasing in price, with ties broken in favor of quantity and submission time. The algorithm traverses the list, accepting offers until supply is reached, and skipping offers that cannot be satisfied from the remaining supply. The algorithm is computationally efficient ($O(m)$, where m is the number of bids).

However, the information provided by the auction makes it difficult for participants to determine bidding strategies, for two reasons. First, rather than announce prices, UBid announces all of the current winning bidders, a list which often contains a hundred or more different bidders. Second, even though the auction employs an ascending rule, a bid that is not winning now could become a winning bid in the future. Consider a case with two objects for sale. The first bidder is agent 1, who bids \$5 for one unit. Then agent 2 bids \$6 each for two units, displacing agent 1. Finally, agent 3 bids \$8 for one unit. The algorithm now decrees that agents 3 and 1 are the winners.

In the worst case, this algorithm can lead to arbitrarily bad allocations. Consider the situation in Table VI. When $p_2 > p_3$, the greedy algorithm will select agent 2's offer first, and we will be unable to satisfy agent 3's offer. If $zp_3 > p_2$, then we will miss $zp_3 - p_2$ surplus. We can make this arbitrarily bad by raising z or p_3.

4.4. *Multidimensional Matching Functions*

Recently, much research attention has been focused on the problem of allocating heterogenous, discrete resources. For the purposes of this paper, we consider only those proposed mechanisms that accept bids on bundles

[10] www.ubid.com

of resources, and we describe their matching and price quote functions. Most are based, in part, on computing the optimal surplus from the bids received, which, when the auction is single-sided, is equivalent to computing the combination of bids that maximizes revenue.

4.4.1. *Generalized Vickrey Auction*

The well-known generalized Vickrey auction (GVA) (MacKie-Mason and Varian, 1994) extends the intuition gained from Vickrey's (1961) original work, and results by Clarke (1971) and Groves (1973) in general allocation problems.

The GVA is a direct revelation mechanism, but we include it here because in many cases agent types can be characterized in the form of offer correspondences. Each agent submits a utility function, and the auction computes the optimal allocation and discriminatory payments. An agent pays (receives) its "social impact," which is defined as the value that the others lose (gain) from the agent's presence. The GVA's incentive compatibility property follows from the fact that an agent's bid determines what resources it gets, but not how much it pays (or receives).

4.4.2. *The RSB Mechanism*

The single-seller auction proposed by Rassenti *et al*. (1982) accepts bids on packages and computes a locally efficient allocation from those bids. The winning agent pays their bids. Price quotes are generated by solving two linear relaxations of the optimization problem. The price quotes are separating in that bids above the upper bound are definitely winning, while bids below the lower bound are definitely losing. An agent whose bid falls between the two bounds cannot determine whether that bid will win or lose.

4.4.3. *Adaptive User Selection Mechanism*

Banks et al. (1989) introduced a single-sided, iterative mechanism, called the adaptive user selection mechanism (AUSM), that does not solve the optimization problem directly. Instead, AUSM announces the current best allocation on a bulletin board. To become part of the current best allocation, a new bid has to offer more than the sum of all of the bids it displaces. Bids that are not part of the best allocation are posted in a standby queue designed to facilitate the coordination of two or more smaller bidders combining on a new bid large enough to displace a larger bidder. In AUSM, winning agents pay the value of their bid.

4.4.4. *BICAP*

The BICAP mechanism (Brewer and Plott, 1996) introduces binary feasibility constraints to the basic combinatorial allocation problem. Bids must obey a beat-the-quote rule. The mechanism computes an optimal allocation of the resources (in the study, the right to use railroad tracks), subject to the specified constraints. CPCA (Brewer, 1999) modifies BICAP in a manner similar to AUSM, relying on the bidders to themselves calculate improving allocations. As an incentive, in CPCA the bidder identifying an improvement gets a share of the improved surplus, with this fraction increasing with time elapsed.

4.4.5. *RAD Mechanism*

Recently, Demartini *et al*. proposed the Resource Allocation Design (RAD) (DeMartini *et al*., 1998), which combines features of AUSM and the simultaneous ascending auction used by the FCC. RAD allows agents to place bids on bundles and then computes the locally efficient allocation. A linear program is solved to generate approximate prices for the individual items, and a beat-the-quote rule is used. Agents need to maintain eligibility by continuing to win items or by submitting new bids.

4.4.6. *i*Bundle

Another single-sided, ascending combinatorial auction is Parke's *i*Bundle (Parkes, 1999). *i*Bundle allows bids on bundles and computes the locally efficient allocation. It associates payments with bundles, in effect, announcing a payment lattice as a price quote. There are three variations of *i*Bundle, which differ based on the manner in which payments are computed: *i*Bundle(2) announces anonymous payments to every agent, *i*Bundle(3) announces discriminatory payments, and *i*Bundle(d) announces discriminatory payments for some agents, and anonymous payments for the rest. Winning agents pay the value of their bids.

4.4.7. *A1BA*

We have recently proposed the ascending k-bundle auction (Wurman, 1999; Wurman and Wellman, 1999), where the parameter $k = 1$. This auction, abbreviated A1BA, is a single-sided, iterative auction that accepts bids on bundles and computes a locally efficient allocation. It differs from AUSM, RAD, and *i*Bundle in that agents are not necessarily charged the value of their bid. Instead, AkBA uses a method for computing uniform payments on bundles that ensures that the final allocation satisfies the conditions of a *payment equilibrium*.[11] This computation is also used to generate (separating) price quotes.

5. PARAMETRIZATION OF WELL-KNOWN AUCTIONS

Many classic and online auctions can be classified using the parameters described herein. Tables VII and VIII present the parameter values for some of the auctions mentioned in this paper. In cases where several common variations exist, such as the CDA, we describe just one of them.

[11] A payment equilibrium is an allocation and set of payments in which no agent would prefer some other bundle to the one it is allocated. The equilibrium concept requires that each agent receive at most one bundle.

TABLE VII
Parameter Choices That Describe Three Online Auctions

	eBay Standard	eBay "Dutch"	UBid "Traditional"
Bidding rules			
Buyers:sellers	{many:1}	{many:1}	{many:1}
Expressiveness	single-unit point	single-price point	single-price point
Bid refinements	NA	divisible	indivisible
Seller bid dominance	none	none	none
Buyer bid dominance	none	none	none
Beat-the-quote	buyer	buyer	buyer
Activity	NA	NA	NA
Information revelation			
Price quotes	ask	ask	nonlinear
Quote timing	activity	activity	activity
Order book	winner	winners	winners
Transaction history	NA	NA	NA
Clearing policy			
Clear timing	fixed time	fixed time	inactivity
Closing conditions	fixed time	fixed time	inactivity
Matching function	$\gamma^{k=1}$	$\gamma^{\kappa=1}$	γ^{greedy}
Tie breaking	earliest	quantity, earliest	quantity, earliest
Auctioneer fees	entrance, nonlinear	entrance, nonlinear	entrance, nonlinear

TABLE VIII
Parameter Choices That Describe Three Classic Auctions

	English Outcry	Vickrey	CDA
Bidding rules			
Buyers:sellers	{many:1}	{many:1}	{many:many}
Expressiveness	single-unit point	single-unit point	single-price point
Bid refinements	NA	NA	indivisible
Seller bid dominance	none	none	none
Buyer bid dominance	none	none	none
Beat-the-quote	buyer	NA	NA
Activity	NA	NA	NA
Information revelation			
Price quotes	ask	none	bid-ask
Quote timing	activity	none	activity
Order book	open	closed	closed
Transaction history	NA	NA	prices
Clearing policy			
Clear timing	inactivity	fixed time	activity
Closing conditions	inactivity	fixed time	never
Matching functions	$\gamma^{k=1}$	$\gamma^{k=0}$	γ^{ET}
Tie breaking	earliest	arbitrary	earliest
Auctioneer fees	none	none	none

6. CONCLUSION

The auction design space presented in this paper captures the essential similarities and differences of many auction mechanisms in a more descriptive and useful format than the traditional taxonomic perspective. This parametrization is not exhaustive, but it is much more extensive than any others we have seen. We have found this organization of auction policy characteristics very useful in our development of the Michigan Internet AuctionBot. In particular, the deconstruction of auctions into functional components with parametrized behaviors coincides nicely with the object-oriented programming approach.

In addition, the parametrization serves as an organizational framework in which to classify research in auction analysis, and uncovers many new, potentially useful, mechanisms. We have seen instances in our own work in which we assumed an agent behavior as part of a protocol—namely, having sellers increase their bids (Walsh and Wellman, 1998)—and later found the behavior was enforceable by a nonobvious combination of rules.

Finally, the parametrization facilitates the communication of auction rules to software agents—a critical step in the development of electronic commerce agents. The AuctionBot agent programming interface includes a method for software agents to request the rules of the auction. Thus, through this single interface, agents can participate in the full range of auction types supported by the server.

ACKNOWLEDGMENTS

This work was supported in part by grants from NSF (National Young Investigator Award IRI-9457624), AFOSR (F49620-97-0175), and DARPA (Information Survivability Program Grant F30602-97-1-0228), and an equipment donation from IBM. We thank Tracy Mullen, Jeffrey MacKie-Mason, Daniel Reeves, and the many students who worked on the AuctionBot project. The second author also acknowledges fruitful discussions on this topic with Yoav Shoham, Eithan Ephrati, Jim Herzog, and auction developers at TradingDynamics, Inc.

REFERENCES

Anderson, A., Birgean, I., and MacKie-Mason, J. K. (1999). "Bilateral Negotiation with Fees," in *IBM/IAC Workshop on Internet-Based Negotiation Technologies*.

Banks, J. S., Ledyard, J. O., and Porter, D. P. (1989). "Allocating Uncertain and Unresponsive Resources: An Experimental Approach," *Rand J. Econ*. **20**, 1–22.

Blackmon, D. A. (2000). "E-Commerce (A Special Report): Revamping the Model—Where the Money Is: Consumer E-Commerce is Sexy, but Business-to-Business May Be a Lot More Profitable," *Wall Street J*. page R30, Apr. 17.

Brewer, P. J. (1999). "Decentralized Computation Procurement and Computational Robustness in a Smart Market," *Econ. Theory* **13**, No. 1, 41–92.

Brewer, P. J., and Plott, C. R. (1996). "A Binary Conflict Ascending Price (BICAP) Mechanism for the Decentralized Allocation of Right to Use Railroad Tracks," *Int. J. Ind. Organization* **14**, 857–886.

Cheng, J. Q., and Wellman, M. P. (1998). "The WALRAS Algorithm: A Convergent Distributed Implementation of General Equilibrium Outcomes," *Comput. Econ*. **12**, 1–24.

Clarke, E. H. (1971). "Multipart Pricing of Public Goods," *Public Choice* **11**, 17–33.

Dalton, G. (1999). "Bazaar Advantages," *Information Week*, pp. 42–48, May 10.

Demange, G., Gale, D., and Sotomayor, M. (1986). "Multi-item Auctions," *J. Polit. Econ*. **94**, 863–872.

DeMartini, C., Kwasnica, A. M., Ledyard, J. O., and Porter, D. (1998). "A New and Improved Design for Multi-object Iterative Auctions," Technical report, California Institute of Technology.

Doorenbos, R. B., Etzioni, O., and Weld, D. S. (1997). "A Scalable Comparison-Shopping Agent for the World-Wide Web," in *First International Conference on Autonomous Agents*, pp. 39–48.

Engelbrecht-Wiggans, R. (1980). "Auctions and Bidding Models: A Survey," *Manag. Sci*. **26**, 119–142.

Friedman, D. (1993). "The Double Auction Market Institution: A Survey," in *The Double Auction Market: Institutions, Theories, and Evidence* (D. Friedman and J. Rust, Eds.), pp. 3–26. Reading, MA: Addison-Wesley.

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

Garey, M. R., and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman.

Groves, T. (1973). "Incentives in Teams," *Econometrica* **41**, 617–631.

Guttman, R. H., Moukas, A. G., and Maes, P. (1998). "Agent-Mediated Electronic Commerce: A Survey," *Knowledge Eng. Rev.* **13**, No. 2, 147–159.

Hu, J., and Wellman, M. P. (1998). "Online Learning about Other Agents in a Dynamic Multiagent System," in *Second International Conference on Autonomous Agents (Agents 98)*, Minneapolis, pp. 239–246.

Kumar, M., and Feldman, S. I. (1998). "Internet Auctions," in *3rd USENIX Workshop on Electronic Commerce*, pp. 49–60.

MacKie-Mason, J. K., and Varian, H. R. (1994). "Generalized Vickrey Auctions," Technical report, University of Michigan.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford: Oxford Univ. Press.

McAfee, R. P. (1992). "A Dominant Strategy Double Auction," *J. Econ. Theory* **56**, 434–450.

McAfee, R. P., and McMillan, J. (1987). "Auctions and Bidding," *J. Econ. Lit.* **25**, 699–738.

McAfee, R. P., and McMillan, J. (1996). "Analyzing the Airwaves Auction," *J. Econ. Perspect.* **10**, No. 1, 159–175.

McCabe, K. A., and Smith, V. L. (1993). "Designing a Uniform-Price Double Auction. An Experimental Evaluation," in *The Double Auction Market: Institutions, Theories, and Evidence* (D. Friedman and J. Rust, Eds.), pp. 307–332. Reading, MA: Addison-Wesley.

McMillan, J. (1994). "Selling Spectrum Rights," *J. Econ. Perspect.* **8**, No. 3, 145–162.

Milgrom, P. (1987). "Auction Theory," in *Advances in Economic Theory: Fifth World Congress* (Truman F. Dewley, Ed.). Cambridge, UK: Cambridge Univ. Press.

Milgrom, P. R., and Weber, R. J. (1982). "A Theory of Auctions and Competitive Bidding," *Econometrica* **50**, 1089–1122.

Mullen, T., and Wellman, M. P. (1996). "Market-Based Negotiation for Digital Library Services," in *Second USENIX Workshop on Electronic Commerce, Oakland, CA*, pp. 259–269.

Parkes, D. C. (1999). "*i*Bundle: An Efficient Ascending Price Bundle Auction," in *Proceedings of the ACM Conference on Electronic Commerce*, pp. 148–157.

Rassenti, S. J., Smith, V. L., and Bulfin, R. L. (1982). "A Combinational Auction Mechanism for Airport Time Slot Allocation," *Bell. J. Econ.* **13**, 402–417.

Rodríguez, J., Noriega, P., Sierra, C., and Padget, J. (1997). "FM96.5, A Java-Based Electronic Auction House," in *Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM '97)*.

Rodríguez-Aguilar, J. A., Martín, F. J., Noriega, P., Garcia, P., and Sierra, C. (1998). "Competitive Scenarios for Heterogenous Trading Agents," in *Second International Conference on Autonomous Agents*, pp. 293–300.

Rothkopf, M. H., Pekeč, A., and Harstad, R. M. (1998). "Computationally Manageable Combinational Auctions," *Manage. Sci.* **44**, No. 8, 1131–1147.

Rust, J., Miller, J. H., and Palmer, R. (1994). "Characterizing Effective Trading Strategies: Insights from a Computerized Double Auction Tournament," *J. Econ. Dynam. Control* **18**, 61–96.

Satterthwaite, M. A., and Williams, S. R. (1989). "Bilateral Trade with the Sealed Bid *k*-Double Auction: Existence and Efficiency," *J. Econ. Theory* **48**, 107–133.

Satterthwaite, M. A., and Williams, S. R. (1993). "The Bayesian Theory of the *k*-Double Auction," in *The Double Auction Market: Institutions, Theories, and Evidence* (D. Friedman and J. Rust, Eds.), pp. 99–124. Reading, MA: Addison-Wesley.

Ungar, L. H., Parkes, D. C., and Foster, D. P. (1998). "Cost and Trust Issues in Online Auctions," in *Agents-98 Workshop on Agent Mediated Electronic Trading, Minneapolis*.

Vickrey, W. (1961). "Counterspeculation, Auctions, and Competitive Sealed Tenders," *J. Finance* **16**, 8–37.

Walsh, W. E., and Wellman, M. P. (1998). "A Market Protocol for Distributed Task Alllocation," in *Third International Conference on Multiagent Systems, Paris*, pp. 325–332.

Wellman, M. P. (1993). "A Market-Oriented Programming Environment and Its Application to Distributed Multicommodity Flow problems," *J. Artificial Intell. Res.* **1**, 1–23.

Wellman, M. P., and Walsh, W. E. (2000). "Distributed Quiescence Detection in Multiagent Negotiation," in *Fourth International Conference on Multiagent Systems, Boston*, pp. 317–324.

Wellman, M. P., and Wurman, P. R. (1998a). "Market-Aware Agents for a Multiagent World," *Robot. Autonomous Syst.* **24**, 115–125.

Wellman, M. P., and Wurman, P. R. (1998b). "Real Time Issues for Internet Auctions," in *IEEE Workshop on Dependable and Real-Time E-Commerce Systems, Denver, CO*.

Wurman, P. R. (1999). *Market Structure and Multidimensional Auction Design for Computational Economies*, Ph.D. thesis, University of Michigan.

Wurman, P. R., Walsh, W. E., and Wellman, M. P. (1998a). "Flexible Double Auctions for Electronic Commerce: Theory and Implementation," *Decision Support Syst.* **24**, 17–27.

Wurman, P. R., and Wellman, M. P. (1999). "Equilibrium Prices in Bundle Auctions," in *AAAI-99 Workshop on Artificial Intelligence for Electronic Commerce*.

Wurman, P. R., Wellman, M. P., and Walsh, W. E. (1998b). "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents," in *Second International Conference on Autonomous Agents, Minneapolis*, pp. 301–308.