

An Iterative Generalized Vickrey Auction: Strategy-Proofness without Complete Revelation

David C. Parkes

Department of Computer and Information Science,
University of Pennsylvania,
200 South 33rd Street,
Philadelphia, PA 19104
dparkes@unagi.cis.upenn.edu

Abstract

The generalized Vickrey auction (GVA) is a strategy-proof combinatorial auction, in which truthful bidding is the optimal strategy for an agent. In this paper we address a fundamental problem with the GVA, which is that it requires agents to compute and reveal their values for all combinations of items. This can be very difficult for bounded-rational agents with limited or costly computation. We propose an experimental design for an *iterative* combinatorial auction. We have a theoretical proof that the auction implements the outcome of the Vickrey auction in special cases, and initial experimental results support our conjecture that the auction implements the outcome of the Vickrey auction in all cases. The auction has better information properties than the sealed-bid GVA: in each round agents must only bid for the set of bundles that *maximize* their utility given current ask prices, which does not require agents to compute their exact values for every bundle.

Introduction

A central problem in distributed open systems with self-interested agents, each with private information, is to compute optimal system-wide solutions to a global problem that involves all the agents. Market-based methods have been proposed to solve problems of this type; e.g., distributed scheduling (Wellman *et al.* 2000), supply-chain (Walsh *et al.* 2000), course registration (Graves *et al.* 1993), airport scheduling (Rassenti *et al.* 1982), and air-conditioning for an office building (Huberman & Clearwater 1995).

The Generalized Vickrey Auction (GVA) (Varian & MacKie-Mason 1995) has received wide attention in the literature, e.g. Wellman *et al.* (2000) and Hunsberger & Grosz (2000), because of its incentive properties; truthful bidding is the optimal strategy for an agent in the GVA. The GVA solves the combinatorial allocation problem (CAP) (Rothkopf *et al.* 1998; de Vries & Vohra 2000), in which there is a discrete set of items to allocate to agents that have values for combinations of items, e.g. "I only want A if I also get B". The goal in the CAP is to compute the allocation that maximizes total value. Expressing contingencies of this kind is important in many applications; for example, to bid for a compatible pair of take-off and landing slots in an airport scheduling problem.

In this paper we address a fundamental shortcoming of the GVA, which is that it requires complete information from agents, i.e. every agent must reveal its value for all pos-

sible combinations of items. This can be very difficult for bounded-rational agents, with limited or costly computation and hard valuation problems. The complete information requirement arises because of the *single-shot* nature of the auction: every agent submits a sealed-bid to the auctioneer from which the allocation is computed. Without an option to ask an agent for more information a mechanism can only compute the efficient allocation in every problem instance with complete information up-front about agents' valuation functions.

In comparison, an *iterative* GVA can terminate with the same outcome (allocation and payments) but with less information revelation. An iterative auction can elicit information from agents dynamically, as required to determine the efficient allocation. Terminating with the Vickrey outcome provides an iterative procedure with much of the same strategy-proofness as the sealed-bid GVA. The design of an iterative GVA is stated as an important open problem in the auction design literature (Bikhchandani & Ostroy 1998; Milgrom 2000). However, iterative Vickrey auctions are only known for special cases (Kelso & Crawford 1982; Demange, Gale, & Sotomayor 1986; Gul & Stacchetti 2000; Ausubel 2000), with restrictions on agent valuation functions.

We propose an experimental design for an ascending-price combinatorial auction. We have a theoretical proof that the auction terminates with the Vickrey outcome in special cases, and initial experimental results support our conjecture that the auction implements the Vickrey outcome in all cases. The auction extends *iBundle* (Parkes 1999a; Parkes & Ungar 2000b) and builds on the ideas introduced in "proxy and adjust" (Parkes & Ungar 2000c). The goal is to implement the Vickrey outcome with best-response agent strategies, i.e. if agents bid in each round for the bundle(s) that maximize their utility.

The auction has two distinct phases. The first phase is used to determine the efficient (value-maximizing) allocation, while the second-phase is used to determine Vickrey payments. However, this transition from Phase I to Phase II is hidden from participants. The auction design is quite novel:

1. First, we adjust agents' payments after the auction terminates, so that agents do not pay their final bid prices. This allows the implementation of non-equilibrium solutions, which is important because the GVA outcome can not al-

ways be supported in equilibrium (Bikchandani & Ostroy 1998).

2. Second, we introduce “dummy agents” during Phase II to force continued bidding from winning agents and to elicit enough information to adjust agent payments towards Vickrey payments.

Our methodology is to construct an auction to implement a *primal-dual* algorithm for the GVA with best-response agent strategies. Best-response bids from agents provide information about the complementary-slackness conditions in a primal-dual formulation, and can be used to adjust towards an optimal solution. Bertsekas (1990) makes a similar connection in his AUCTION algorithm for the simple assignment problem.

Outline

In the next section we provide a brief discussion of the consequences of agent bounded-rationality on mechanism design. We also present an application of a combinatorial auction with hard valuation problems, the *package delivery problem*. We define the GVA, and show by example how it is possible to compute the outcome of the auction without complete value information. We discuss two methods to reduce information revelation in the GVA: bidding programs, and dynamic methods. We then describe our new auction procedure, and illustrate the mechanism on a couple of examples. We present initial experimental results that confirm our conjecture that the auction implements the outcome of the GVA. Finally, we present a brief theoretical justification for the design of the auction. We close with conclusions and suggestions for future work.

Mechanism Design with Bounded-Rational Agents

We will begin with a brief discussion of the implications of *agent* bounded-rationality on mechanism design. Nisan & Ronen (2000) provide a good introduction to related concerns that follow from the computational requirements placed on the *mechanism infrastructure*.

One very important consideration is the amount of value information that is required by a mechanism. Single-shot mechanisms, such as the GVA, can fail when it is too expensive for agents to compute and reveal their value for every combination of items. We describe an example below in which agents have hard valuation problems. In comparison, an iterative mechanism can solve realistic problems with less information, eliciting only the minimal information required to compute an efficient allocation.

In addition to solving problems without complete information revelation it is also important that agents can participate without *computing* exact values for all bundles. It is not enough for an auction to require less information from agents if the agents must still compute values for all bundles to provide that information.

This focus on information revelation conflicts with a central result in mechanism design, the “revelation principle” (Green & Laffont 1977; Myerson 1981). The revelation

principle states that in the design of mechanisms we can restrict attention to “direct revelation” mechanisms which request complete information about an agent’s valuation function. However, the revelation principle assumes unlimited computational resources, both for agents in submitting valuation functions, and for the auctioneer in computing the outcome of a mechanism (Ledyard 1989).

Before we continue, it is interesting (and perhaps surprising) to note that there is one sense in which agent bounded-rationality can *help* in mechanism design. Perhaps we can design mechanisms that cannot be manipulated unless an agent can solve an NP-hard computational problem. Nisan & Ronen (2000) follow this line of reasoning, and describe the concept of “feasible truthfulness” in mechanism design.

The Combinatorial Allocation Problem

In this paper we focus on mechanisms for the Combinatorial Allocation Problem (CAP), which is a quite general resource allocation problem in which agents have values for combinations of items. Many resource allocation problems can be modeled as a CAP, including the job-shop scheduling, course registration and airport scheduling problems mentioned in the introduction.

Formally, let \mathcal{G} represent a set of discrete items to allocate to \mathcal{I} agents. Each agent has a value $v_i(S) \geq 0$ for bundles $S \subseteq \mathcal{G}$ of items. The CAP is to allocate items to maximize total value over all agents:

$$\begin{aligned} \max_S \sum_{i \in \mathcal{I}} v_i(S_i) & \quad (\text{CAP}) \\ \text{s.t. } S_i \cap S_j = \emptyset, \quad \forall i, j \in \mathcal{I}, i \neq j \end{aligned}$$

It is well known that the CAP is NP-hard, it is equivalent to the maximum weighted set packing problem (Rothkopf *et al.* 1998).

However, this straightforward statement of complexity hides another very important computational problem in the CAP, which arises in the case that a hard *computational problem* must be solved by each agent to compute its value for each bundle. This is quite likely in many applications, for example whenever an agent must solve a local optimization problem to compute its value for different outcomes. As an example, consider the *package delivery problem*.

Example: The package delivery problem. The package delivery problem is an example of a distributed optimization problem in which agents have hard valuation problems.

The problem can be modeled as a combinatorial allocation problem: items \mathcal{G} represent packages for delivery, with pick-up and delivery requirements (e.g. locations, times, priority, etc.); agents \mathcal{I} represent individual delivery companies, each with its own transportation resources, e.g. vans, airplanes, trucks, etc.

An agent’s value, $v_i(S)$, for bundle of packages $S \subseteq \mathcal{G}$ is computed as the *payment* it receives for delivering the package minus its cost. Assume that the payment is a linear sum of payments $p(x)$ for packages $x \in S$, and denote cost $C_i(S)$.

$$v_i(S) = \sum_{x \in S} p(x) - C_i(S)$$

The global objective in the package delivery problem, captured by this CAP formulation, is to determine an allocation that maximizes the total value across all agents. This is equivalent to allocating packages at minimal cost and dropping packages that cannot be delivered profitably.

It is reasonable to expect the cost to an agent for the pick-up and delivery of a particular package to depend on factors such as: prior commitments (e.g. prescheduled pick-ups and deliveries), van capacities, delivery locations and times. For example, the additional cost to deliver a package is much less if a van is already scheduled to make a pick-up at the destination location. Formally, the cost for a particular set of packages (i.e. a commitment to perform a set of deliveries) might perhaps be computed as a *multi-vehicle capacity-constrained traveling salesperson problem*, which is NP-hard.

This example should give a sense in which value information can be hard to provide in distributed optimization problems.

The GVA: Incentive-Compatibility with Complete Revelation

In this section we define the generalized Vickrey auction (GVA), which is an incentive-compatible and efficient mechanism for the combinatorial allocation problem. We observe that the GVA requires complete information revelation from agents, and demonstrate that it is in fact often possible to compute the outcome of the GVA (payments and allocation) with less information from each agent.

We describe two approaches to reduce information revelation but still compute the outcome of the GVA: (1) allow agents to submit *bidding programs*, which the auctioneer can query to determine values for particular bundles; (2) involve agents *dynamically* during the execution of an algorithm to compute the Vickrey outcome, and request additional information on-the-fly as required. We state a number of drawbacks with bidding programs, and suggest that iterative auctions are a useful class of dynamic methods.

The Generalized Vickrey Auction

Each agent $i \in \mathcal{I}$ submits a (possibly untruthful) valuation function, \hat{v}_i , to the auctioneer. The auctioneer solves the CAP with these reported values, computing allocation $S^* = (S_1^*, \dots, S_I^*)$ with value V^* . This is the allocation implemented by the auction.

The auctioneer also solves CAP without each agent in turn, computing the best allocation $(S_{-i})^*$ without agent i for value $(V_{-i})^*$. The Vickrey payment to agent i is computed as:

$$p_{\text{vick}}(i) = (V_{-i})^* - \sum_{j \neq i} \hat{v}_j(S_j^*)$$

In words, an agent pays the marginal negative effect that its participation has on the (reported) value of the other agents. Equivalently, the Vickrey payment can be computed as a discount $\Delta_{\text{vick}}(i)$ from its bid price, $\hat{v}_i(S_i^*)$, i.e. $p_{\text{vick}}(i) = \hat{v}_i(S_i^*) - \Delta_{\text{vick}}(i)$, for Vickrey discount:

$$\Delta_{\text{vick}}(i) = V^* - (V_{-i})^*$$

This interpretation is more consistent with the method to compute Vickrey payments in the iterative GVA that we introduce later in the paper.

It is quite straightforward to show that the optimal strategy of an agent in the GVA is to bid its true valuation function, $\hat{v}_i = v_i$, whatever the bids of other agents.

Theorem 1. (Groves 1973) Truthful bidding is a dominant strategy in the GVA.

Furthermore, because $V^* \geq (V_{-i})^*$ the discount is always non-negative and agents pay less than their bids, while the discount is not so large that the adjusted price is ever negative, because $(V_{-i})^* \geq V^* - \hat{v}_i(S_i^*)$ implies that $\Delta_{\text{vick}}(i) \leq \hat{v}_i(S_i^*)$.

Computational Problems

The GVA has unreasonable computational properties in many interesting problems.

Problem 1. *The GVA is intractable for the auctioneer.* Once the auctioneer has received bids from each agent it must solve multiple CAP instances, once to compute V^* , and once to compute $(V_{-i})^*$ for every agent i in the optimal allocation.

Problem 2. *The GVA is intractable for agents with hard valuation problems.* The GVA requires complete information revelation from each agent. As soon as an agent submits approximate or missing information: (i) there is some probability that the agent will do worse—in terms of its value for the bundle it receives and/or the price it pays—than if it had revealed complete and accurate information; and (ii) the allocation implemented in the GVA might not be efficient.

The first problem has received some attention, but only in the context of sealed-bid auctions, and without addressing the second problem. In general, introducing approximation algorithms for winner-determination can break the incentive-compatibility of the GVA (Kfir-Dahav, Monderer, & Tennenholtz 1998; Nisan & Ronen 2000). In comparison, the problem of information revelation in the GVA has received little attention. One exception is the discussion of bidding programs in Nisan (2000).

Solving the GVA without Complete Information

In this section we demonstrate how it is possible to *compute* and *verify* the optimality of a solution to the GVA without complete information about agents' values. We ignore for the moment the question of *how* to elicit the required information, and simply demonstrate that we can compute solutions with incomplete information.

Example 1. Single-item auction with 3 agents, and values $v_1 = 16, v_2 = 10, v_3 = 4$. The Vickrey outcome is to sell the item to agent 1 for agent 2's value, i.e. for 10. Instead of information $\{v_1, v_2, v_3\}$ it is sufficient to know $\{v_1 \geq 10, v_2 = 10, v_3 \leq 10\}$ to compute this outcome.

Example 2. Consider the simple combinatorial allocation problem instance in Table 1, with items A, B and agents 1, 2, 3. The

	<i>A</i>	<i>B</i>	<i>AB</i>
Agent 1	0	<i>a</i>	<i>b</i>
Agent 2	10	0	10
Agent 3	0	0	15

Table 1: Agent values in Example 2.

values of agent 1 for item *B* and bundle *AB* are stated as $a \leq b$ and $b \leq 15$, but otherwise left undefined. Consider the following cases:

[$a < 5$] In this case the GVA assigns bundle *AB* to agent 3, with $V^* = 15$, $(V_{-3})^* = \max[10 + a, b]$, so that the payment for agent 3 is $p_{\text{vick}}(3) = 15 - (15 - \max[10 + a, b]) = \max[10 + a, b]$. It is sufficient to know $\{a \leq 5, b \leq 15, \max[10 + a, b]\}$ to compute the outcome.

[$a \geq 5$] In this case the GVA assigns item *B* to agent 1 and item *A* to agent 2, with $V^* = 10 + a$, $(V_{-1})^* = 15$, and $(V_{-2})^* = 15$. The payment for agent 1 is $p_{\text{vick}}(1) = a - (10 + a - 15) = 5$ and the payment for agent 2 is $p_{\text{vick}}(2) = 10 - (10 + a - 15) = 15 - a$. It is sufficient to know $\{a, b \leq 15\}$ to compute the outcome.

Note that it is not necessary to compute the *value* of the optimal allocation \mathbf{S}^* ; we only need to compute the allocation to each agent. Consider Example 1. We can compute the optimal allocation (give the item to agent 1) with information $v_1 \geq \{v_2, v_3\}$, and without knowing the exact value of v_1 .

Also, it is not even necessary to compute V^* and $(V_{-i})^*$ to compute Vickrey payments because common terms cancel. In Example 1, it is enough to know the value of v_2 to compute agent 1's Vickrey payment because the value of v_1 cancels: $p_{\text{vick}}(1) = v_1 - \Delta_{\text{vick}}(1) = v_1 - (v_1 - v_2) = v_2$.

Methods to Reduce Information Revelation

We will consider two different methodologies to reduce information revelation but compute the GVA outcome: (1) bidding programs, and (2) dynamic methods.

Bidding Programs. Instead of requiring agents to compute and reveal their valuation functions we might ask agents to provide a program (or *oracle*) that can be used by the auctioneer to compute values for bundles on demand (Nisan 2000). This will reduce agent computation if it is easier for an agent to construct the program than it is to compute its explicit value for all bundles.

However, this approach may not be very useful in practice for a number of reasons: (1) *trust*, agents might be reluctant to reveal all the information that goes into assigning a value to an outcome; (2) *cost of constructing the program*, it might be costly to collect all necessary information to define such an autonomous program; (3) *computational burden*, this merely shifts computation to the auctioneer.

Note, in particular, that if the only functionality provided by the bidding program is to compute the exact value, $v_i(S)$, for bundle *S* and agent *i*, the auctioneer must now compute the complete value of every agent for all possible bundles to compute the GVA outcome. One can make a straightforward information theoretic argument that the value of every

agent for every bundle must be at least *considered* to compute the solution to a CAP instance. More usefully, the agent might provide a program that can give approximate value information to the auctioneer, or that allows richer query-response modes.

Dynamic Methods. An alternative approach is to “open up” the algorithm for computing the outcome of the GVA, and involve agents dynamically in the computational process.

The algorithm might ask agents for the following types of information during its execution:

- Ordinal information, i.e. “which bundle has highest value out of S_1, S_2 and S_3 ?”
- Approximate information, i.e. “is your value for bundle S_1 greater than 100?”
- Best-response information, i.e. “which bundle do you want at prices $p(S)$?”
- Equivalence-set information, i.e. “is there an item that is substitutable for *A*?”

Notice that in all of these cases an agent can respond without computing its exact value for all bundles.

Iterative price-directed auctions provide a useful class of dynamic methods to implement mechanisms.

First, iterative auctions can solve realistic problems without complete information from agents, and without agents computing their exact values for all bundles. Consider the classic English auction, which is an ascending-price auction for a single item. It is sufficient to determine in each round whether two or more agents have value greater than the ask price; it is *not* necessary to know the exact value of every agent for the item.¹

Second, iterative auctions are quite *transparent* methods to implement the outcome of a mechanism. For example, the information requested from agents in each round of the English auction is captured by a best-response to the current ask prices, i.e. bidding for the item while the price is less than the agent's value.

Third, agents can follow myopic best-response strategies in iterative price-directed auctions without computing exact values for all bundles. For example, an agent can follow a best-response bidding strategy in a price-directed iterative auction with lower and upper bounds on its values for bundles. Myopic best-response only requires that an agent bids for the bundle(s) with maximum utility (value - price) in each round. This utility-maximizing set of bundles can be computed by refining the values on individual bundles until the utility of one or more bundles dominates all other bundles.²

¹In earlier work, Parkes (1999b) demonstrates that iterative auctions can compute better solutions than single-shot auctions for a simple model of agent bounded-rationality.

²Standard algorithmic approaches can provide lower and upper bounds on values; e.g. *anytime algorithms* such as heuristic search compute lower-bounds on optimal solutions to maximization problems, while introducing *problem relaxations* and solving easy special-cases can compute upper-bounds on value.

An Iterative Generalized Vickrey Auction

In this section we describe our experimental ascending-price combinatorial auction. We have a theoretical proof that the auction implements the outcome of the GVA in special cases, and experimental results support our conjecture that the auction implements the Vickrey outcome in all cases. This is quite significant, because an iterative Vickrey auction inherits much of the strategy-proofness of the GVA but with less information revelation from agents.

Although hidden from participants, the auction has two distinct phases. Phase I is identical to *iBundle*, an ascending-price combinatorial auction introduced in Parkes (1999a). Phase I ends when *iBundle* terminates, at which point the auctioneer stores the provisional allocation and computes *initial price discounts*, using method *Adjust** introduced in Parkes & Ungar (2000c). The allocation at the end of Phase I is implemented at the end of the auction. Then, the auctioneer uses a simple test to determine whether to execute Phase II, or terminate the auction immediately.

The purpose of Phase II is to collect enough additional information to be able to compute Vickrey payments. At the end of Phase II, or after Phase I if Phase II is skipped, the agents' payments are computed as their bid prices at the end of Phase I minus the total discounts computed over both phases.

We will first describe the theoretical properties of the auction, both proven properties and conjectures. Then, we describe the auction in some detail, and illustrate it with a couple of worked examples.

Auction Properties

Let us first *assume* that agents follow a truthful myopic best-response bidding strategy:

Definition 1. A *truthful myopic best-response bidding strategy* is to bid to maximize utility in the current round, taking prices as fixed.

We will later claim that the auction implements the Vickrey outcome with truthful myopic best-response, which justifies this assumption by Theorems 4 and 5 (below).

The first result is that the allocation implemented in the auction is efficient, at least for small enough bid increments. This result follows immediately from the efficiency of *iBundle* with best-response strategies (Parkes & Ungar 2000b), because Phase I is identical to *iBundle* and the final allocation is the allocation as computed at the end of Phase I.

Theorem 2. (Parkes & Ungar 2000b) *The auction terminates with an allocation within $3 \min\{|\mathcal{G}|, |\mathcal{I}|\}\epsilon$ of the optimal solution, for myopic best-response bidding strategies.*

where $|\mathcal{G}|$ is the number of items, $|\mathcal{I}|$ is the number of agents, and ϵ is the *minimal bid increment* in the auction, the rate at which prices are increased (see below). Clearly as $\epsilon \rightarrow 0$ the auction terminates with the optimal allocation.

The following unproved conjecture states that the auction

also implements Vickrey payments, and is therefore an iterative GVA:

Proposition 1. *The auction terminates with the outcome of the GVA for agents that follow a myopic best-response bidding strategy, as the minimal bid increment $\epsilon \rightarrow 0$.*

Initial experimental results support this conjecture. Our belief in this statement also follows from the theoretical underpinnings of the auction design, which we discuss later in the paper.

The auction provably implements the outcome of the GVA in the same cases that *iBundle* with *Adjust** computes the Vickrey outcome. These special cases include:

Theorem 3. (Parkes & Ungar 2000c) *The auction terminates with the outcome of the GVA in the following special cases: assignment problem with unit-demands; with multiple identical items and subadditive valuation functions (i.e. decreasing returns); with linear-additive valuation functions in items.*

From Proposition 1, one might think that myopic best-response should be a dominant strategy in the iterative auction, because truthful bidding is a dominant strategy in the GVA. In fact manipulation remains possible (if perhaps difficult), with some strategy other than best-response.

We can make the following statement about an iterative auction that *myopically implements* the outcome of the GVA; i.e. an auction that implements the outcome of the GVA with agents that follow truthful myopic best-response strategies.

Theorem 4. (Gul & Stacchetti 2000) *Truthful myopic bidding is a sequentially rational best-response to truthful myopic bidding by other agents in an iterative auction A that myopically-implements the Vickrey outcome.*

The proof follows quite directly from the incentive-compatibility of the GVA. Basically, for any other strategy the agent selects a GVA outcome for some non-truthful valuation function, which is less preferable than the GVA outcome for its true valuation function.

To make a stronger claim about strategy-proofness we must somehow restrict the strategies that agents can follow. In particular, it would be useful to restrict agents to follow a (possibly untruthful) best-response strategy (i.e. best-response for some $v' \neq v$):

Theorem 5. *Truthful myopic bidding is a dominant strategy in an iterative auction A that myopically-implements the Vickrey outcome, if all agents are constrained to following a (possibly untruthful) best-response bidding strategy.*

We pursued this idea in Parkes & Ungar (2000c), with the idea of “proxy bidding agents”. A proxy bidding agent can sit between a participant and the auction and make only best-response bids, based on value information provided by participants. Crucially, participants can provide incremental information; proxy agents do not require complete information from agents. All that is required is that there is enough information to compute a best-response to current

prices. The proxy agent checks that the information is consistent over the course of the auction.

We can make the following claim about the strategy-proofness of a proxied iterative Vickrey auction:

Theorem 6. *Truthful dynamic information revelation is a sequentially rational best-response to truthful dynamic information revelation by other agents in an iterative auction A with best-response proxy bidding agents that myopically implements the GVA.*

Unfortunately, it is still not quite true that truthful incremental information revelation to proxy agents is a dominant strategy. This is because agents do not commit to a single valuation function at the start of the auction, but instead report incremental value information during the auction. As long as the information is self-consistent, an agent can condition the values it chooses to report based on bids from other agents during the auction. There remains a slight “gap” in strategy-proofness that, theoretically at least, an agent might be able to squeeze through.

Auction Description

The auction is an ascending-price combinatorial auction in which agents can bid for bundles of items. The auctioneer increases prices on bundles as bids are received and computes a provisional allocation in each round of the auction to maximize revenue.

There are two distinct phases: Phase I, in which the final allocation is determined, followed by Phase II, in which final payments are determined. Intermediate computation performed at the end of Phase I determines whether or not Phase II can be safely skipped.

Both phases follow the price update rules, bidding rules, and winner-determination rules of *iBundle(3)* (Parkes 1999a), which is the version of *iBundle* with separate ask prices for each agent throughout the auction. The termination condition in Phase II, and the additional steps performed at the end of each round in Phase II, are new.

Common Elements. First, let us describe the elements that are common to both phases:

- **Bids.** Agents can place exclusive-or bids for bundles, e.g. S_1 XOR S_2 , to indicate that an agent wants either all items in S_1 or all items in S_2 but not both S_1 and S_2 . Each bid is associated with a *bid price*, $p_{\text{bid},i}(S)$, from agent i for bundle S . The bid price must either be within ϵ of, or greater than, the *ask price*, $p_{\text{ask},i}(S)$, announced by the auctioneer. Parameter $\epsilon > 0$ is the *minimal bid increment*, the minimal price increase in the auction (see below). Agents must repeat bids for bundles in the provisional allocation, but otherwise are only committed to bids for a single round.
- **Winner-determination.** The auctioneer computes a *provisional allocation* in each round, to maximize revenue given agents’ bids.
- **Price-update.** The auctioneer maintains separate *ask prices*, $p_{\text{ask},i}(S)$, for each agent. The prices on bundles to agent i are increased in any round that the agent bids

unsuccessfully, i.e. in any round that the agent does not receive a bundle in the provisional allocation.

For every bundle S that received an unsuccessful bid, the ask price in the next round is increased to $\max[p_{\text{bid},i}(S) + \epsilon, p_{\text{ask},i}(S)]$, for minimal bid increment ϵ . The initial ask prices are zero.

Only bundles that have received unsuccessful bids are explicitly priced. However, an additional consistency rule states that the ask price on all bundles is at least as high as the greatest ask price of any bundle they contain, i.e. $p_{\text{ask}}(S') \geq p_{\text{ask}}(S)$ for all $S \subset S'$.

Phase I. The termination condition at the end of Phase I is as follows:

- **Phase I termination.** Phase I terminates when: [T1] all agents submit the same bids in two consecutive rounds, or [T2] all agents that bid receive a bundle.

This is the end of Phase I. Let $\mathbf{S}^* = (S_1^*, \dots, S_I^*)$ denote the allocation at the end of Phase I, P^* denote the auctioneer’s revenue, $W^* \subseteq \mathcal{I}$ denote the set of agents that receive a bundle in \mathbf{S}^* , and $\mathbf{p}_{\text{bid}}^* = (p_{\text{bid},1}^*, \dots, p_{\text{bid},I}^*)$ denote the final bid price of each agent for the bundle it receives in the final allocation, i.e. $p_{\text{bid},i}^* = p_{\text{bid},i}(S_i^*)$.

Intermediate Computation. At the end of Phase I the auctioneer performs some intermediate computation to determine: (i) initial discounts to each agent, and (ii) whether to enter Phase II or terminate immediately.

Let $\Delta_{\text{init}}(i) \geq 0$ denote the initial discount, computed at this stage, and $\Delta_{\text{extra}}(i) \geq 0$ denote any additional discount computed during Phase II. An agent’s final payment at the end of the auction is its discounted bid price at the end of Phase I: $\max[p_{\text{bid},i}^* - \Delta_{\text{init}}(i) - \Delta_{\text{extra}}(i), 0]$.

We need the following definitions:

Definition 2. Let $\text{MAXREV}(P_{-i})$ denote the *second-best revenue-maximizing allocation to the auctioneer at the current ask prices*; the *revenue-maximizing allocation over all allocations without agent i* , i.e. $\text{MAXREV}(P_{-i}) = \max_{\mathbf{S}} \sum_{j \neq i} p_{\text{ask},j}(S_j)$ for feasible allocations \mathbf{S} with $S_i = \emptyset$. Also, let $(W_{-i})^* \subseteq (\mathcal{I} \setminus i)$ denote the set of agents that receive a bundle in this allocation.

Definition 3. The *dependents* $\alpha(i)$ of agent i are:

$$\alpha(i) = W^* \setminus ((W_{-i})^* \cup i), \text{ if } i \in W^*$$

$$\alpha(i) = \emptyset, \text{ otherwise.}$$

In other words, the dependents of agent i are agents that receive a bundle in allocation \mathbf{S}^* but do not receive a bundle in the revenue-maximizing allocation without agent i at the current ask prices.

Definition 4. The set $\text{ACTIVE} \subseteq \mathcal{I}$ of agents is the set of *active agents*, all agents that are dependent of some agent at the current ask prices; i.e. $i \in \text{ACTIVE} \Rightarrow \exists j \text{ s.t. } i \in \alpha(j)$.

At the end of Phase I the auctioneer computes initial discounts, the sets of dependents, and determines the active

agents:

1. Compute initial discount

$$\Delta_{\text{init}}(i) = P^* - \text{MAXREV}(P_{-i})$$

for every agent i in allocation \mathbf{S}^* , or $\Delta_{\text{init}}(i) = 0$ otherwise.

2. Compute dependents $\alpha(i)$ for every agent $i \in W^*$. If $\Delta_{\text{init}}(i) \geq p_{\text{bid},i}^*$ then set $\alpha(i) = \emptyset$ (i.e. remove all dependents for the agent). Initialize the set of *ACTIVE* agents.

The following simple test is performed to determine whether or not to enter Phase II.³

- *Early Termination Test.* Phase II is skipped if there are no active agents.

If there is at least one active agent then more information is required to compute Vickrey payments, and the auction continues into Phase II.

Phase II. The purpose of Phase II is to compute an additional discount $\Delta_{\text{extra}}(i) \geq 0$ to each agent with dependents.

We introduce *dummy agents* to compete with agents in the allocation at the end of Phase I. The auctioneer simulates the dummy agents, generating bids in each round.

We say that an agent *participates* in the auction if it is either in the provisional allocation or bidding at least the ask price for one or more bundles. Similarly, we say that an agent has *dropped out* of the auction if it neither in the provisional allocation or submitting new bids.

At the start of Phase II the auctioneer initializes $\Delta_{\text{extra}}(i) = 0$ for all agents, and introduces a dummy agent for any agent that dropped out of the auction in the last round of Phase I.

A simple method is used to construct dummy agents:

Definition 5. *The valuation function of a dummy agent for agent j is based on the ask prices of agent j : set $v(S) = p_{\text{ask},j}(S) + L$ for bundles S with $p_{\text{ask},j}(S) > 0$, and $v(S) = 0$ for all other bundles, for some large constant $L > 0$.*⁴

The auctioneer performs the following additional steps at the end of each round of Phase II:

1. Update the dependents of agents. For example, if $\alpha(i) \neq \emptyset$ then compute $\text{MAXREV}(P_{-i})$ and $(W_{-i})^*$, restricting the allocation to all real agents except agent i . (i.e. without any dummy agents). Update $\alpha(i)$ based on W^* from the end of Phase I and $(W_{-i})^*$.
2. For each agent $i \in W^*$ and with $\alpha(i) \neq \emptyset$, increment $\Delta_{\text{extra}}(i)$ by $\sum_{j \in \alpha(i)} \Delta_{\text{incr}}(j)$ where $\Delta_{\text{incr}}(j) \geq 0$ is the increase in bid price by agent j for bundle S_j^* since the previous round.
3. If $\Delta_{\text{init}}(i) + \Delta_{\text{extra}}(i) \geq p_i^*$, then set $\alpha(i) = \emptyset$
4. Update the set of *ACTIVE* agents.

³This is a reinterpretation of the Vickrey Test in Parkes & Ungar (2000c).

⁴The valuation function will satisfy “free disposal” ($v(S) \geq v(S')$ for all $S' \subset S$) because of the price consistency maintained across ask prices.

5. Introduce new dummy agents. First, for any agent that has just dropped out of the auction.⁵ Second, if we detect a state of *quiescence* for the active agents.⁶ In this case we use a simple rule to choose the agent to receive a dummy agent; select (1) an agent with no dummy that is not active; or failing that (2) an active agent with no dummy; or failing that (3) an active agent that already has at least one dummy agent.

The termination condition is:

- *Phase II termination.* Terminate Phase II when all active agents have dropped out of the auction, or when the set of *ACTIVE* agents is empty.

Final Price Adjustment. Finally, when the auction terminates: allocation \mathbf{S}^* as computed at the end of Phase I is implemented; and agent payments

$$p_i = \max[0, p_{\text{bid},i}^* - \Delta_{\text{init}}(i) - \Delta_{\text{extra}}(i)]$$

are computed.

Worked Examples

We will illustrate the auction on Example 2, for different values of a and b .

Case ($a = b = 3$). *Phase I:* $\mathbf{S}^* = (\emptyset, \emptyset, AB)$, $P^* = 13$, $W^* = \{3\}$, $p_{\text{bid}}^* = (0, 0, 13)$. *Intermediate Computation.* $(\mathbf{S}_{-3})^* = (B, A, \emptyset)$, $\text{MAXREV}(P_{-3}) = 13$, $(W_{-3})^* = \{1, 2\}$, $\alpha(3) = \emptyset$, $\Delta_{\text{init}}(3) = 13 - 13 = 0$. *ACTIVE* = \emptyset . *Skip Phase II.* *Outcome:* Allocate bundle AB to agent 3 for $p_3 = 13 - (0 + 0) = 13$. This is the Vickrey payment: $p_{\text{vick}}(3) = 15 - (15 - 13) = 13$.

Case ($a = b = 10$). *Phase I:* $\mathbf{S}^* = (A, B, \emptyset)$, $P^* = 15$, $W^* = \{1, 2\}$, $p_{\text{bid}}^* = (8, 7, 0)$. *Intermediate Computation.* $(\mathbf{S}_{-1})^* = (\emptyset, \emptyset, AB)$, $\text{MAXREV}(P_{-1}) = 15$, $(W_{-1})^* = \{3\}$, $\alpha(1) = \{1, 2\} \setminus \{3, 1\} = \{2\}$, $\Delta_{\text{init}}(1) = 15 - 15 = 0$, $(\mathbf{S}_{-2})^* = (\emptyset, \emptyset, AB)$, $\text{MAXREV}(P_{-2}) = 15$, $(W_{-2})^* = \{3\}$, $\alpha(2) = \{1, 2\} \setminus \{3, 2\} = \{1\}$, $\Delta_{\text{init}}(2) = 15 - 15 = 0$. *ACTIVE* = $\{1, 2\}$.

Phase II. Introduce a dummy agent for agent 3, with values $v_4 = (0, 0, 15 + L)$ for a large $L > 0$. As prices increase agent 1 drops out first, when $p_1(B) > 10$. At this time $\Delta_{\text{extra}}(2) = 2$ because agent 1’s bid has increased by 2 since the end of Phase I. A dummy agent is introduced for agent 1, with values $v_5 = (0, 10 + L, 10 + L)$. Finally, agent 2 drops out when $p_2(A) > 10$, at which time $\Delta_{\text{extra}}(1) = 3$ because agent 2’s bid has increased by 3 since the end of Phase I.

Outcome: Allocate item B to agent 1 for $p_1 = 8 - (0 + 3) = 5$ and item A to agent 2 for $p_2 = 7 - (0 + 2) = 5$. These are the Vickrey payments: $p_{\text{gva}}(1) = p_{\text{gva}}(2) = 10 - (20 - 15) = 5$.

Case ($a = b = 20$). *Phase I and Intermediate Computation* is the same as in case $a = b = 10$. *Phase II.* Introduce a dummy agent for agent 3 with values $v_4 = (0, 0, 15 + L)$ for a large $L > 0$.

⁵If a dummy agent already exists for this agent, then replace with this new one.

⁶The precise definition of quiescence is not too important. We consider that the auction is in quiescence if: (1) the same active agents have participated in the auction for the past three rounds; and (2) all participating active agents have been allocated the same (non-empty) bundle in the provisional allocation in the past three rounds, and for the same price.

As prices increase agent 2 drops out first, when $p_2(A) > 10$ and $\Delta_{\text{extra}}(1) = 3$. Introduce a dummy agent for agent 2 with value $v_5 = (10 + L, 0, 10 + L)$. Finally, agent 1 enters $(S_{-2})^*$, when $p_1(B) = 15$ and $\Delta_{\text{extra}}(2) = 7$. At this stage $\alpha(2) = \emptyset$ and agent 1 is no longer active.

Outcome: Allocate item B to agent 1 for $p_1 = 8 - (0 + 3) = 5$ and item A to agent 2 for $p_2 = 7 - (0 + 7) = 0$. These are the Vickrey payments: $p_{\text{gva}}(1) = 20 - (30 - 15) = 5$ and $p_{\text{gva}}(2) = 10 - (30 - 20) = 0$.

Experimental Analysis

In this section we describe initial experimental results, which support our conjecture that the ascending-price combinatorial auction computes the outcome of the GVA.

The auction is tested in a suite of problem instances: problems PS 1–12 from (Parkes 1999a), and also problems Decay, Weighted-random, Random and Uniform in Sandholm (1999).⁷ Each problem set defines a distribution over agents' values for bundles of items.

We measure the distance between agent payments in the auction and GVA payments with an L_2 norm, as $[\sum_i (p_i - p_{\text{gva}}(i))^2]^{1/2}$. For a particular bid increment we compute the average distance to Vickrey payments over the instances in which the auction terminates with the optimal allocation. As the bid increment gets small this fraction approaches 100%, and therefore the proportion of trials over which we measure the distance to Vickrey payments approaches 100%.⁸

Figure 1 plots the distance between Vickrey payments and auction payments against the “correctness” of the auction, i.e. the fraction of instances in which the allocation at the end of Phase I is the efficient allocation. As we reduce the minimal bid increment correctness increases and we move to the right in the plots. The corresponding allocative efficiency increases from around 90% at 23% correctness, to almost 100% at correctness of 65% and above.

We plot the distance to GVA payments at the end of Phase I, after the initial price discounts at the end of Phase I, and at the end of Phase II. For comparison, we also compute the average distance between minimal competitive equilibrium (CE) prices and Vickrey payments (see the next section for a discussion of the relevance of minimal CE prices).⁹

⁷Problem sizes (num agents, num items, num bundles) are set to (8, 50, 120) in Decay, (20, 50, 400) in Uniform, (8, 30, 80) in Random, and (15, 50, 300) in Weighted Random. Decay parameter $\alpha = 0.85$. The results are averaged over 40 trials.

⁸This provides a more useful measure of distance than computing the average L_2 distance over all trials, including those in which the wrong agents are in the final allocation. As an alternative we might compute distance with an L_1 norm over all trials; this is equivalent to comparing the total revenue in the auction with the revenue in the GVA.

⁹Additional statistics for PS 1-12: at 95% correctness the average number of rounds in Phase I is 149, compared to 18 in Phase II; each round in Phase I takes an average of 0.5s, compared to 2.8s in Phase II; the agent valuation information provided at the end of Phase I is 77%, and increases to 83% at the end of Phase II, using a metric introduced in Parkes (1999a); and approximately 1-in-3

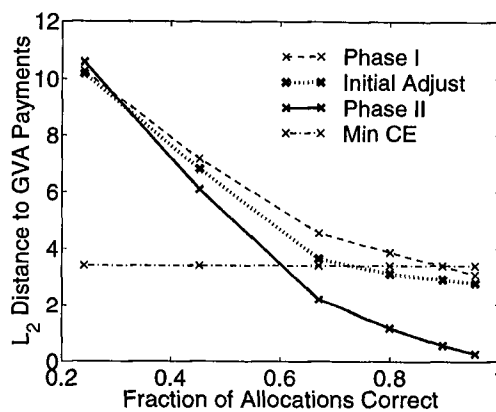


Figure 1: Distance to Vickrey Payments in PS 1–12.

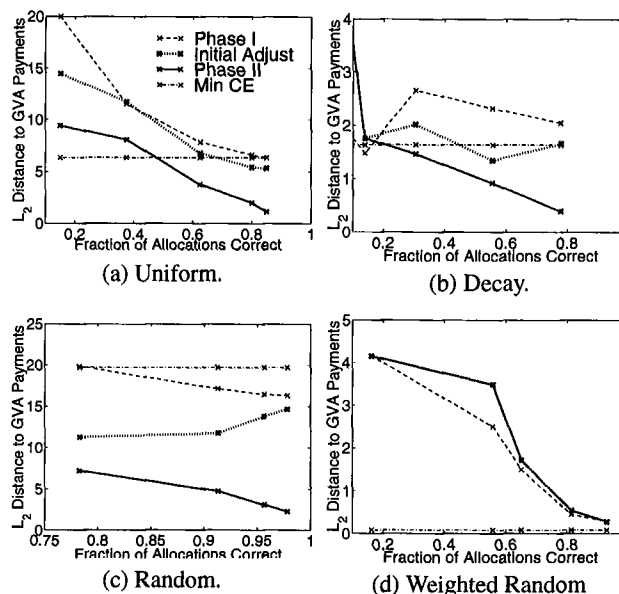


Figure 2: Distance to Vickrey Payments.

Figure 2 illustrates the performance of the auction in problems Uniform, Decay, Random, and Weighted Random. In all problems allocative efficiency approaches 100% for small bid increments, and the distance to GVA payments approaches zero.

Payments in the auction converge to GVA payments in all problems that we examined, as the minimal bid increment gets small and the correctness of the allocation in the auction approaches 100%. The effect of Phase II is quite significant, while the initial price adjustment at the end of Phase I has a smaller effect.

It is also worth noting that the auction implements the Vickrey outcome even in problems in which the outcome is *not* supported in any competitive equilibrium; notice that the distance between the minimal CE prices and the GVA payments is non-zero in all experiments.

agents receive a dummy agent during Phase II.

Auction Design: Theoretical Motivation

In this section we briefly provide some theoretical justification for the design of Phase I and Phase II of the auction.

Phase I. In Phase I we determine the efficient allocation, based on myopic best-response bids from agents. Phase I is equivalent to *iBundle*, which implements a primal-dual algorithm for the CAP with best-response bids from agents, with respect to linear program formulations $LR(\mathcal{E}_3)$ and $DLR(\mathcal{E}_3)$ introduced by Bikchandani & Ostroy (1998).

In each round the provisional allocation is a feasible primal solution and the ask prices are a feasible dual solution. The complementary-slackness conditions, which are necessary and sufficient for optimality of the provisional allocation, can be given interpretations in the auction. In Parkes & Ungar (2000b) we prove that *iBundle* terminates with prices and an allocation that satisfy complementary-slackness conditions, and therefore with an efficient allocation.

Intermediate Computation. The intermediate computation determines the initial price discounts to each agent. In Parkes & Ungar (2000c) we demonstrate that given competitive-equilibrium prices (p_1, \dots, p_I) , prices $(p_1 - \Delta_{\text{init},1}, p_2, \dots, p_I)$ are also in competitive-equilibrium (CE). *CE prices are any set of prices that satisfy complementary slackness with the optimal primal solution, i.e. any optimal dual solution.* Note that they are not unique.

Bikchandani & Ostroy (1998) prove that CE prices are *upper-bounds* on Vickrey payments, therefore adjusted price $p_{\text{bid},i}^* - \Delta_{\text{init},i}$ is an upper-bound on an agent's Vickrey payment. This is the adjusted payment computed in Intermediate Computation at the end of Phase I of the auction.

In addition, we prove in Parkes & Ungar (2000c) (see Theorem 9) that a sufficient condition for the adjusted payments to equal Vickrey payments is that there are no *active* agents; i.e. that all agents in the allocation at the end of Phase I are also in the revenue-maximizing allocation without any single agent.

Phase II. The motivation for Phase II is to force active agents to bid higher prices for bundles received in the optimal allocation. The ask prices to all agents remain valid competitive equilibrium prices during Phase II. Therefore, we can still compute discount $\Delta(i) = P^* - \text{MAXREV}(P_{-i})$ for agent i as the auction continues. This is computed dynamically in the auction, as the sum of $\Delta_{\text{init}}(i)$ and $\Delta_{\text{extra}}(i)$.

In Parkes & Ungar (2000c) we prove necessary and sufficient conditions for this discount to compute Vickrey payments:

- The ask prices must leave every agent not in the optimal allocation indifferent between receiving no bundle and any bundle it receives in a second-best allocation.
- The ask prices must leave every agent in the optimal allocation indifferent between the bundle it receives in the optimal allocation and any bundle it receives in a second-best allocation.
- Every agent in the final allocation must either: (a) have an adjusted payment of zero; or (b) the ask price to all

dependents of that agent for the bundle they receive in the final allocation must equal their value for the bundle.

It is not hard to show that these conditions are all satisfied (to within ϵ) at the end of Phase II. We make agents in W^* but not in $(W_{-i})^*$ continue to bid higher prices, and increase their ask prices, until they enter $(W_{-i})^*$ or drop out of the auction.

The tricky part of the proof that we terminate with GVA payments is to show that the competition from the dummy agents is sufficient to make Phase II terminate, i.e. to push up the bid prices high enough of all active agents. We believe that the current protocol for detecting quiescence during Phase II and introducing new dummy agents is sufficient for this purpose.

Example.

For example, consider case $(a = b = 10)$ in Example 2. At the end of Phase I we have the following CE prices: $\mathbf{p}_{\text{ask},1} = (0, 8, 8)$, $\mathbf{p}_{\text{ask},2} = (7, 0, 7)$, and $\mathbf{p}_{\text{ask},3} = (0, 0, 15)$. During Phase II, agents 1 and 2 bid continue to bid and face higher ask prices. At the end of Phase II, we have the following additional sets of CE prices: (i) $\mathbf{p}_{\text{ask},1} = (0, 10, 10)$, $\mathbf{p}_{\text{ask},2} = (7, 0, 7)$, and $\mathbf{p}_{\text{ask},3} = (0, 0, 15)$, from which we can compute minimal CE prices $\mathbf{p}_{\text{ask},1} = (0, 10, 10)$, $\mathbf{p}_{\text{ask},2} = (5, 0, 5)$, and $\mathbf{p}_{\text{ask},3} = (0, 0, 15)$; and (ii) $\mathbf{p}_{\text{ask},1} = (0, 8, 8)$, $\mathbf{p}_{\text{ask},2} = (10, 0, 10)$, and $\mathbf{p}_{\text{ask},3} = (0, 0, 15)$, from which we can compute minimal CE prices $\mathbf{p}_{\text{ask},1} = (0, 5, 5)$, $\mathbf{p}_{\text{ask},2} = (10, 0, 10)$, and $\mathbf{p}_{\text{ask},3} = (0, 0, 15)$. The first set of minimal CE prices supports the Vickrey payment to agent 2. The second set of minimal CE prices supports the Vickrey payment to agent 1.

Discussion

It is important that agents cannot identify the transition from Phase I to Phase II of the auction because nothing that an agent bids in Phase II will change either the final allocation or its own final payment. The only effect of an agent's bids in Phase II is to reduce the final payment made by other agents. If it is costly to participate in the auction an agent would choose to drop out after Phase I. In addition, there are opportunities for collusion between agents in Phase II (just as the GVA itself is vulnerable to collusion).

Certainly, we must hide bids from dummy agents in Phase I (or give the dummy agents false identities). Each agent only needs information about its own ask prices, and whether or not it is receiving a bundle in the provisional allocation. Agents do not need any information about the bids, prices, or allocations of other participants. We must also be sure that agents cannot distinguish the competitive effects of bids from dummy agents from the competitive effects of bids from real agents. This is our reasoning for constructing dummy agents to "clone" the real agents that compete for items in Phase I of the auction.

Future Work

There are a number of interesting areas for future work. First, it should be possible to reduce the computational demands on the auctioneer. For example, it would be useful to allow the auctioneer to recompute the dependent agents

in each round of Phase II without explicitly computing the second-best revenue-maximizing allocations.

Second, we would like to reduce the level of price discrimination in the auction. For example, in application to the allocation of a single item, the current auction maintains a separate ask price for each agent. In comparison, the English auction implements the Vickrey outcome with a single ask price which is the same to all agents.

Looking ahead, proxy bidding agents, situated between real agents and the auctioneer, suggest a method to integrate rich preference semantics into the auction; e.g. with ordinal, cardinal, and constraint-based information. This is an interesting area for future work.

Conclusions

In this paper we have addressed the unreasonable information demands that the generalized Vickrey auction can place on agents. We introduce an experimental design for an iterative combinatorial auction. The auction provably terminates with the outcome of the GVA in special cases, and experimental results support our hypothesis that the auction implements the outcome of the GVA in all cases. While preliminary, the auction may lead to practical and strategy-proof mechanisms with many interesting applications.

Acknowledgments

This research was funded in part by National Science Foundation Grant SBR 97-08965. In addition, I gratefully acknowledge financial support from an IBM Fellowship.

References

- Ausubel, L. 2000. An efficient dynamic auction for heterogeneous commodities. Technical report, Department of Economics, University of Maryland.
- Bertsekas, D. P. 1990. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces* 20(4):133–149.
- Bikhchandani, S., and Ostroy, J. M. 1998. The package assignment model. Technical report, Anderson School of Management and Department of Economics, UCLA.
- de Vries, S., and Vohra, R. 2000. Combinatorial auctions: A brief survey. Technical report, MEDS Department, Kellogg Graduate School of Management, Northwestern University.
- Demange, G.; Gale, D.; and Sotomayor, M. 1986. Multi-item auctions. *Journal of Political Economy* 94(4):863–872.
- Graves, R. L.; Schrage, L.; and Sankaran, J. 1993. An auction method for course registration. *Interfaces* 23(5):81–92.
- Green, J., and Laffont, J. J. 1977. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica* 45:427–438.
- Groves, T. 1973. Incentives in teams. *Econometrica* 41:617–631.
- Gul, F., and Stacchetti, E. 2000. The English auction with differentiated commodities. *Journal of Economic Theory* 66–95.
- Huberman, B. A., and Clearwater, S. 1995. A multi-agent system for controlling building environments. In *Proc. 1st International Conference on Multi-Agent Systems (ICMAS-95)*, 171–176.
- Hunsberger, L., and Grosz, B. J. 2000. A combinatorial auction for collaborative planning. In *Proc. 4th International Conference on Multi-Agent Systems (ICMAS-00)*.
- Kelso, A. S., and Crawford, V. P. 1982. Job matching, coalition formation, and gross substitutes. *Econometrica* 50:1483–1504.
- Kfir-Dahav, N. E.; Monderer, D.; and Tennenholtz, M. 1998. Mechanism design for resource bounded agents. Technical report, Technion.
- Ledyard, J. O. 1989. Incentive compatibility. In Eatwell, J.; Milgate, M.; and Newman, P., eds., *Allocation, Information and Markets*. W. W. Norton. 141–151.
- Milgrom, P. 2000. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy* 108. forthcoming.
- Myerson, R. B. 1981. Optimal auction design. *Mathematics of Operation Research* 6:58–73.
- Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*.
- Nisan, N., and Ronen, A. 2000. Computationally feasible VCG mechanisms. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*.
- Parkes, D. C. 1999a. iBundle: An efficient ascending price bundle auction. In *Proc. 1st ACM Conf. on Electronic Commerce (EC-99)*, 148–157.
- Parkes, D. C. 1999b. Optimal auction design for agents with hard valuation problems. In *Proc. IJCAI-99 Workshop on Agent Mediated Electronic Commerce*. Stockholm.
- Parkes, D. C., and Ungar, L. H. 2000b. Iterative combinatorial auctions: Theory and practice. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, 74–81.
- Parkes, D. C., and Ungar, L. H. 2000c. Preventing strategic manipulation in iterative auctions: Proxy agents and price-adjustment. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, 82–89.
- Rassenti, S. J.; Smith, V. L.; and Bulfin, R. L. 1982. A combinatorial mechanism for airport time slot allocation. *Bell Journal of Economics* 13:402–417.
- Rothkopf, M. H.; Pekeč, A.; and Harstad, R. M. 1998. Computationally manageable combinatorial auctions. *Management Science* 44(8):1131–1147.
- Sandholm, T. 1999. An algorithm for optimal winner determination in combinatorial auctions. In *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 542–547.
- Varian, H., and MacKie-Mason, J. K. 1995. Generalized Vickrey auctions. Technical report, University of Michigan.
- Walsh, W.; Wellman, M.; and Ygge, F. 2000. Combinatorial auctions for supply chain formation. In *Proc. ACM Conference on Electronic Commerce*, 260–269.
- Wellman, M. P.; Walsh, W. E.; Wurman, P. R.; and MacKie-Mason, J. K. 2000. Auction protocols for decentralized scheduling. *Games and Economic Behavior*. To appear.