Journal of Artificial Intelligence Research 33 (2008) 33–77 Submitted 09/07; published 09/08

ICE: An Expressive Iterative Combinatorial Exchange

Benjamin Lubin BLUBIN@EECS.HARVARD.EDU
Adam I. Juda ADAMJUDA@POST.HARVARD.EDU
Ruggiero Cavallo CAVALLO@EECS.HARVARD.EDU
Sébastien Lahaie SLAHAIE@EECS.HARVARD.EDU
Jeffrey Shneidman JEFFSH@EECS.HARVARD.EDU
David C. Parkes PARKES@EECS.HARVARD.EDU
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138

Abstract

We present the design and analysis of the first fully expressive, iterative combinatorial exchange (ICE). The exchange incorporates a tree-based bidding language (*TBBL*) that is concise and expressive for CEs. Bidders specify lower and upper bounds in *TBBL* on their value for different trades and refine these bounds across rounds. These bounds allow price discovery and useful preference elicitation in early rounds, and allow termination with an efficient trade despite partial information on bidder valuations. All computation in the exchange is carefully optimized to exploit the structure of the bid-trees and to avoid enumerating trades. A proxied interpretation of a revealed-preference activity rule, coupled with simple linear prices, ensures progress across rounds. The exchange is fully implemented, and we give results demonstrating several aspects of its scalability and economic properties with simulated bidding strategies.

1. Introduction

Combinatorial exchanges combine and generalize two different mechanisms: double auctions and combinatorial auctions. In a double auction (DA), multiple buyers and sellers trade units of an identical good (McAfee, 1992). In a combinatorial auction (CA), a single seller has multiple heterogeneous items up for sale (de Vries & Vohra, 2003; Cramton, Shoham, & Steinberg, 2006). Each buyer in a CA may have complementarities ("I want A and B") or substitutabilities ("I want A or B") between goods, and is provided with an expressive bidding language to describe these preferences. A common goal in the design of both DAs and CAs is to implement the efficient allocation, which is the allocation that maximizes total social welfare.

A combinatorial exchange (CE) (Parkes, Kalagnanam, & Eso, 2001) is a combinatorial double auction that brings together multiple buyers and sellers to trade multiple heterogeneous goods. CEs have potential use in wireless spectrum allocation (Cramton, Kwerel, & Williams, 1998; Kwerel & Williams, 2002), airport takeoff and landing slot allocation (Ball, Donohue, & Hoffman, 2006; Vossen & Ball, 2006), and in financial markets (Saatcioglu, Stallaert, & Whinston, 2001). In all of these domains there are incumbents with property rights, and it is necessary to facilitate a complex multi-way reallocation of resources. Another potential application domain for CEs is to allocate resources in shared distributed systems, such as PlanetLab (Fu, Chase, Chun, Schwab, & Vahdat, 2003). CEs can also find use in task allocation in robot teams, making them a potentially powerful tool to the multi-agent systems community (Gerkey & Mataric, 2002; Bererton, Gordon, & Thrun, 2003; Dias, Zlot, Kalra, & Stentz, 2006). Finally, CEs have promise as mechanisms for expressive sourcing by multiple bid-takers, perhaps representing different profit centers within an organization; see the associated work on expressive sourcing using one-sided CAs (Sandholm, 2007).

This paper presents the design of the first *fully expressive, iterative* combinatorial exchange (ICE). In designing an iterative exchange, we share the motivation of earlier work on iterative CAs: we wish to mitigate elicitation costs by focusing bidders, in this case through price discovery and activity rules, on their values for relevant trades. This is important because determining the value on even a single potential trade can be a challenging problem in complex domains (Sandholm & Boutilier, 2006; Compte & Jehiel, 2007). Moreover, bidders often wish to reveal as little information as possible to avoid leaking information to competitors. In describing the central *design principles* that support the ICE mechanism, we highlight the following aspects:

- A bidder interacts with ICE by first defining a structured representation of his valuation for different trades. Defined in the tree-based bidding language (*TBBL*), this concisely defines the set of trades of interest to the bidder. The bidder must annotate the tree with initial lower and upper bounds on his value for different trades.

- Having lower and upper bounds on valuations allows the exchange to identify both a provisional trade and provisional payments in each round, and to generate a provisional clearing price on each item in the market. In each round of ICE, each bidder is required to tighten the bounds on his *TBBL* bid so as to make precise which trade is most preferred given the current prices.

- ICE is a hybrid between a demand-revealing process and a direct-revelation mechanism, with simple (linear) prices guiding preference elicitation but bids submitted through direct claims about valuation functions in the *TBBL* language, and these expressive bids being finally used to clear the exchange.

When ICE terminates, a payment rule is used to determine the payments made, and received, by each participant. While suggesting that these payments be defined in a way that seeks to mitigate opportunities for manipulation in the exchange, ICE is agnostic to the particular payment rule that is adopted. For a given rule, the prices that are quoted in each round are defined in part to approximate these payments, when aggregated across the provisional trade suggested for a bidder. For concreteness, we adopt the *Threshold* rule (Parkes et al., 2001) in defining final payments, which minimizes the *ex post* regret for truthful bidding across all budget-balanced payment rules, when holding the bids from other participants fixed; see also the work of Milgrom (2007). [1] This is not to say that

1. We are not aware of the existence of mechanism design solutions for approximately efficient, but truthful (i.e., with truthful bidding as a dominant-strategy equilibrium) and budget-balanced, sealed bid (i.e., non iterative) CEs. Nevertheless, it is true that any payment rules that are developed can be leveraged directly within ICE and would allow ICE to inherit truthful bidding (i.e., revising *TBBL* bounds to remain consistent with a bidder's true valuation) in an *ex post* Nash equilibrium, just as can be achieved in iterative Generalized Vickrey auctions (Mishra & Parkes, 2007).

incentive issues related to payment rules are not important in the design of successful CEs. Rather, this is just orthogonal to the design of ICE and not the main focus of our work. We do propose novel activity rules, which are themselves designed to mitigate opportunities for strategic behavior.

We highlight the following *technical contributions* made in this work:

- The tree-based bidding language (*TBBL*) extends earlier CA bidding languages to support bidders that wish to simultaneously buy and sell, the specification of valuation bounds, and the use of generalized 'choose' operators to provide more concise representations than OR* and $\mathcal{L}_{GB}$ (Boutilier & Hoos, 2001; Nisan, 2006). *TBBL* can be directly encoded within a mixed-integer programming (MIP) formulation of the winner determination problem.

- Despite quoting prices on items and not bundles of items, ICE is able to converge to the efficient trade with straightforward (i.e., non-strategic) bidders. Efficiency is established through duality theory when prices are sufficiently accurate. Otherwise, a direct proof based on reasoning about the upper and lower valuation bounds is always available, even when the combinatorics of the instance preclude a duality-based proof.

- Preference elicitation is performed through the combination of two novel *activity rules*. The first is a *modified revealed-preference activity rule* (MRPAR), and requires each bidder to make precise which trade is most preferred in each round. The second is a *delta improvement activity rule* (DIAR), and requires each bidder to refine his bid to improve price accuracy or prove that no improvement is possible. When coupled together these rules ensure that useful progress towards determining the efficient trade is made in each round.

To summarize, there are three main reasons to prefer explicit value representations over repeated demand reports in the context of an iterative CE: (a) a provisional allocation can be computed from round 1, since both upper and lower bounds on value are available, (b) the combinatorics of the domain can be directly handled in clearing the exchange and efficiency is not limited by adopting simple (linear) prices, (c) proofs of (approximate) efficiency are available by reasoning directly with bounds on valuations and despite adopting simple (linear) prices.

The exchange is fully implemented in Java (with a C-based MIP solver). We present scalability results showing performance across a wide number of bidders, goods and valuation complexity as well as benchmarks that provide a qualitative understanding of the characteristics of our mechanism. Our experimental results (with straightforward bidders) show that the exchange quickly converges to the efficient trade, taking an average of only 7 rounds for an example domain with 100 goods of 20 different types and 8 bidders with valuation functions containing an average of 112 *TBBL* nodes. In this same domain, we find that bidders can leave upwards of 62% of their maximum attainable value undefined when the efficient trade is known, and 56% once final payments are determined, indicating that bidders are able to leave large amounts of their value space unrefined. The exchange terminates on these problems in an average of 8.5 minutes on a 3.2GHz dual-processor dual-core workstation with 8GB of memory. This includes the time for all winner determination, pricing, and activity rules, as well as the time to simulate agent bidding strategies.

1.1 Related Work

Many ascending-price one-sided CAs are known in the literature (Parkes & Ungar, 2000a; Wurman & Wellman, 2000; Ausubel & Milgrom, 2002; de Vries, Schummer, & Vohra, 2007; Mishra & Parkes, 2007). Direct elicitation approaches, in which bidders respond to explicit queries about their valuations, have also been proposed for one-sided CAs (Conen & Sandholm, 2001; Hudson & Sandholm, 2004; Lahaie & Parkes, 2004; Lahaie, Constantin, & Parkes, 2005). Of particular relevance here are the ascending CAs that are designed to work with simple prices on items (Dunford, Hoffman, Menon, Sultana, & Wilson, 2003; Kwasnica, Ledyard, Porter, & DeMartini, 2005). In computing (approximately competitive) linear prices, we generalize and extend these methods. Building on the work of Rassenti, Smith, and Bulfin (1982), these earlier papers consider bids on bundles individually, and find prices that are exact on winning bids and minimize the pricing error to losing bids. Generalizing to the *TBBL* expressive language, we propose instead to compute prices that minimize the worst-case pricing error over all *bidders* (rather than bids on individual trades), considering the most preferred trade consistent with the *TBBL* bid of each bidder. As in the work of Dunford et al. (2003) and Kwasnica et al. (2005) we incorporate additional tie-breaking stages, in our case to lexicographically minimize the error and then to find prices that closely approximate the provisional *payments*. This latter step appears to be novel.

Linear prices are important in practical applications. Such prices are adopted by the FCC in their wireless spectrum auctions (Cramton, 2006), within clock auctions for the procurement of electricity generation (Cramton, 2003), and are an essential part of the proposed design for an airport landing slot auction at Laguardia airport (Ball et al. 2007). Linear competitive equilibrium prices exist in two-sided markets with indivisibilities for the assignment problem in which each agent will buy or sell a single item (but may be interested in multiple different items) (Shapley & Shubik, 1972). But in general linear, competitive equilibrium prices will not exist in combinatorial markets with nonconvexities; see the work of Kelso and Crawford (1982), Bikhchandani and Mamer (1997), Bikhchandani and Ostroy (2002), and O'Neill, Sotkiewicz, Hobbs, Rothkopf, and Stewart (2005) for related discussions.

ICE has a "proxied" architecture in the sense that bidders submit and refine bounds on *TBBL* bids directly to the exchange, with this information used to drive price dynamics and ultimately to clear the exchange. Earlier work has considered proxied approaches, but in application to one-sided ascending-price CAs (Parkes & Ungar, 2000b; Ausubel & Milgrom, 2002). Given its focus on simple, linear prices, ICE can be considered to provide a two-sided generalization of the *clock-proxy* design of Ausubel, Cramton, and Milgrom, which has an initial stage of linear price discovery followed by a "best-and-final" sealed-bid stage (Ausubel et al., 2006). Activity rules have been shown to be very important in practice. For instance, the Milgrom-Wilson activity rule that requires a bidder to be active on a minimum percentage of the quantity of the spectrum for which it is eligible to bid is a critical component of the auction rules used by the FCC for wireless spectrum auctions (Milgrom, 2004). ICE adopts a variation on the clock-proxy auction's revealed-preference activity rule.

It is well known that exact efficiency together with budget balance is not possible because of the Myerson-Satterthwaite impossibility result (Myerson & Satterthwaite, 1983). Given this, Parkes et al. study sealed-bid combinatorial exchanges and introduced the Threshold payment rule (Parkes et al., 2001); see the work of Milgrom (2007) and Day and Raghavan (2007) for a recent discussion. Double auctions in which truthful bidding is in a dominant strategy equilibrium are known for unit demand settings (McAfee, 1992) and also for slightly more expressive domains (Babaioff & Walsh, 2005; Chu & Shen, 2007). However, no truthful, budget-balanced mechanisms with useful efficiency properties are known for the general CE problem.

Voucher-based schemes have been proposed as an alternative method to extend one-sided CAs to exchanges (Kwerel & Williams, 2002). Such mechanisms collect all goods from sellers and then run a one-sided auction in which sellers can "buy-back" their own goods with vouchers used to provide a seller with a share of the revenue collected on their own goods. Although voucher-based schemes can facilitate the design of exchanges through one-sided auction technology, the ICE design offers the nice advantage of providing equal and symmetric expressiveness to all participants. We are not aware of any previous studies of fully expressive iterative CEs. Smith, Sandholm, and Simmons previously studied iterative CEs, but handle only limited expressiveness and adopt a direct-query based approach with an enumerative internal data structure that does not scale (Smith et al., 2002). A novel feature in their earlier design (not supported here) is *item discovery*, where the items available to trade need not be known in advance. Earlier work has also considered *sealed-bid* combinatorial exchanges for the purpose of contingent trades in financial markets, including aspects of expressiveness and winner determination (Saatcioglu et al., 2001).

Several bidding languages for CAs have previously been proposed, arguably the most compelling of which allow bidders to explicitly represent the logical structure of their valuation over goods via standard logical operators. We refer to these as "logical bidding languages" (Nisan, 2006). Closest in generality to *TBBL* is the $\mathcal{L}_{GB}$ language (Boutilier & Hoos, 2001), which allows for arbitrarily nested levels, combining goods and trades by the standard propositional logic operators, and also provides a *k-of* operator, used to represent a willingness to pay for any k trades it quantifies over; see also the work of Rothkopf, Pekeč, and Harstad (1998) for a restricted tree-based bidding language. In a key insight, Boutilier specifies a MIP formulation for Winner Determination (WD) using $\mathcal{L}_{GB}$, and provides positive empirical performance results using a commercial solver, suggesting the computational feasibility of moving to this more expressive logical language (Boutilier, 2002). *TBBL* shares some structural elements with the $\mathcal{L}_{GB}$ language but has important differences in its semantics. In $\mathcal{L}_{GB}$, the semantics are those of propositional logic, with the same items in an allocation able to satisfy a tree in multiple places. Although this can make $\mathcal{L}_{GB}$ especially concise in some settings, the semantics that we propose provide representational locality, so that the value of one component in a tree can be understood independently from the rest of the tree.

1.2 Outline

Section 2 introduces preliminary concepts, defining the efficient trade and competitive equilibrium prices. Section 3 defines a sealed-bid CE, introducing *TBBL* and providing the MIP that is used to solve winner determination. Section 4 extends *TBBL* to allow for valuation bounds and defines the MRPAR and DIAR activity rules. The main theoretical results are also described as well as our method to determine price feedback in each round. Section 5 gives a number of illustrative examples of the operation of ICE. Section 6 presents our main experimental results. We conclude in Section 7. The Appendix provides an algorithm for each of the two activity rules together with details on the bidding logic used by simulated bidding agents.

2. Preliminaries

The basic environment considers a set of bidders, $N = \{1, \ldots, n\}$, who are interested in trading multiple units of distinct, indivisible goods, where the set of different types of goods is denoted $G = \{1, \ldots, m\}$. Each bidder has an initial endowment of goods and a valuation for different trades. Let $x^0 = (x^0_1, \ldots, x^0_n)$ denote the initial endowment of goods, with $x^0_i = (x^0_{i1}, \ldots, x^0_{im})$ and $x^0_{ij} \in \mathbb{Z}_+$ to indicate the number of units of good type $j \in G$ initially held by bidder $i \in N$. A *trade* $\lambda = (\lambda_1, \ldots, \lambda_n)$ denotes the *change* in allocation, with $\lambda_i = (\lambda_{i1}, \ldots, \lambda_{im})$ and $\lambda_{ij} \in \mathbb{Z}$ denoting the change in the number of units of item j to bidder i. Let $M = \sum_{i\in N} \sum_{j\in G} x^0_{ij}$ denote the total supply in the exchange. We write $i \in \lambda$ to denote that bidder i is *active* in the trade, i.e., buys or sells at least one item.

2.1 The Efficient Trade

Each bidder has a value $v_i(\lambda_i) \in \mathbb{R}$ for his component of trade λ. This value can be positive or negative, and represents the *change in value* between the final allocation $x^0_i + \lambda_i$ and the initial allocation x^0_i. The valuation and initial allocation information is private to each bidder, and we assume that there are no externalities, so that each bidder's value depends only on his individual trade. We assume *free disposal*, so that $v_i(\lambda'_i) \geq v_i(\lambda_i)$ for trade $\lambda'_i \geq \lambda_i$, i.e., for which $\lambda'_{ij} \geq \lambda_{ij}$ for all j. Let $v(\lambda) = \sum_i v_i(\lambda_i)$.

Utility is modeled as quasi-linear, with $u_i(\lambda_i, p) = v_i(\lambda_i) - p$ for trade λ_i and payment $p \in \mathbb{R}$. This implies that bidders are modeled as being risk neutral and assumes that there are no budget constraints. The payment, p, can be negative, indicating the bidder may receive a payment for the trade. We use the term *payoff* interchangeably with *utility*. Because of quasi-linearity, any Pareto optimal (i.e., efficient) trade will maximize the social welfare, which is equivalent to the total increase in value to all bidders due to the trade. Given an instance of the CE problem, defined by tuple (v, x^0), i.e., a valuation profile $v = (v_1, \ldots, v_n)$ and an initial allocation $x^0 = (x^0_1, \ldots, x^0_n)$, the efficient trade λ^*, is defined as follows:

Definition 1 *Given CE instance (v, x^0), the efficient trade λ^* solves*

$$\max_{(\lambda_1,\ldots,\lambda_n)} \sum_i v_i(\lambda_i) \tag{1}$$

$$\text{s.t.} \quad \lambda_{ij} + x^0_{ij} \geq 0, \quad \forall i, \forall j \tag{2}$$

$$\sum_i \lambda_{ij} = 0, \quad \forall j \tag{3}$$

$$\lambda_{ij} \in \mathbb{Z}$$

Constraints (2) ensure that no bidder sells more items than he has in his initial allocation. By free disposal, we can impose strict balance in the supply and demand of goods at the solution in constraints (3), i.e., we can allocate unwanted items to any bidder. We adopt $\mathcal{F}(x^0)$ to denote the set of *feasible trades*, given these constraints and given an initial allocation x^0, and $\mathcal{F}_i(x^0)$ for the set of feasible trades to bidder i. Note that valuation function v_i cannot be explicitly represented as a value for each possible trade to bidder i, because the number of such trades scales as $O(s^m)$, where s is the maximal number of units of any item in the market and there are m different items. The *TBBL* language (introduced in Section 3) leads to a concise formulation of the efficient trade problem as a mixed-integer program.

The initial allocation x_i^0 may be private to agent i. We assume throughout that bidders are truthful in revealing this information, which we motivate by supposing that participants cannot sell items that they do not actually own (or pay a suitably high penalty if they do).

2.2 Competitive Equilibrium Prices

Linear prices, $\pi = (\pi_1, \ldots, \pi_m)$, define a price π_j on each good so that the price to bidder i on a trade λ is defined as $p^\pi(\lambda_i) = \sum_j \lambda_{ij}\pi_j = \lambda_i \cdot \pi$. Such prices play an important role in ICE. Of particular interest is the set of competitive equilibrium prices:

Definition 2 *Linear prices π are competitive equilibrium (EQ) prices for CE problem (v, x^0) if there is some feasible trade $\lambda \in \mathcal{F}(x^0)$ such that:*

$$v_i(\lambda_i) - p^\pi(\lambda_i) \geq v_i(\lambda'_i) - p^\pi(\lambda'_i), \quad \forall \lambda'_i \in \mathcal{F}_i(x^0), \tag{4}$$

for every bidder i. We say that such a trade, λ, is ***supported*** *by prices π.*

Theorem 1 *(Bikhchandani & Ostroy, 2002) Any trade λ supported by competitive equilibrium prices π is an efficient trade.*

In practice, exact EQ prices are unlikely to exist. Instead, it is useful to define the concept of *approximate* EQ prices and an approximately efficient trade:

Definition 3 *Linear prices π are δ-approximate competitive equilibrium (EQ) prices for CE problem (v, x^0) and $\delta \in \mathbb{R}_{\geq 0}$, if there is some feasible trade $\lambda \in \mathcal{F}(x^0)$ such that:*

$$v_i(\lambda_i) - p^\pi(\lambda_i) + \delta \geq v_i(\lambda'_i) - p^\pi(\lambda'_i), \quad \forall \lambda'_i \in \mathcal{F}_i(x^0), \tag{5}$$

for every bidder i.

At δ-approximate EQ prices, there is some trade for which every bidder i is within $\delta \geq 0$ of maximizing his utility. Furthermore, say that trade λ is *z-approximate* if the total value of the trade is within z of the total value of the efficient trade.

Theorem 2 *Any trade λ supported by δ-approximate EQ prices π is a $2\min(M, \frac{n}{2})\delta$-approximate efficient trade.*

Proof: Fix instance (v, x^0) and consider (λ, π). For any trade $\lambda' \neq \lambda$ we have

$$\sum_{i \in \lambda \cup \lambda'} [v_i(\lambda_i) - p^\pi(\lambda_i) + \delta] \geq \sum_{i \in \lambda \cup \lambda'} [v_i(\lambda'_i) - p^\pi(\lambda'_i)], \tag{6}$$

by δ-EQ prices and because values and prices are zero for bidders that do not participate in a trade. We have $\sum_{i\in\lambda\cup\lambda'} p^{\pi}(\lambda_i) = \sum_{i\in\lambda\cup\lambda'} p^{\pi}(\lambda'_i) = 0$ (since $\sum_i p^{\pi}\lambda''_i = \sum_i \lambda''_i \cdot \pi = \sum_i \sum_j \lambda''_{ij}\pi_j = \sum_j \pi_j \sum_i \lambda''_{ij} = 0$, with $\sum_i \lambda''_{ij} = 0$ for all j, for all $\lambda''_i \in \mathcal{F}(x^0)$). Then, $\sum_i v_i(\lambda_i) + \sum_{i\in\lambda\cup\lambda'} \delta_i \geq \sum_i v_i(\lambda'_i)$. Fix $\lambda' := \lambda^*$, for efficient trade λ^*. Then, $\sum_i v_i(\lambda_i) + \Delta \geq \sum_i v_i(\lambda^*_i)$, where

$$\Delta = \sum_{i\in\lambda\cup\lambda'} \delta_i \leq \min(2A^{\#}(x^0), n)\delta \leq \min(2\min(M,n), n)\delta = 2\min(M, \frac{n}{2})\delta \qquad (7)$$

Here $A^{\#}(x^0)$ is the maximal number of bidders that can trade in a feasible trade given x^0. The second inequality follows because no more bidders can trade than there are number of goods to trade or bidders in the market and thus $A^{\#}(x^0) \leq \min(M, n)$. □

3. Step One: A *TBBL*-Based Sealed-Bid Combinatorial Exchange

We first flesh out the details for a non-iterative, *TBBL*-based CE in which each bidder submits a sealed bid in the *TBBL* language.

Bidding language. The tree-based bidding language (*TBBL*) is designed to be expressive and concise, to be entirely symmetric with respect to buyers and sellers, and to easily provide for bidders that are both buying and selling goods; i.e., ranging from simple swaps to highly complex trades. Bids are expressed as annotated *bid trees*, and define a bidder's *change in value* for all possible trades. The main feature of *TBBL* is that it has a general "interval-choose" logical operator on internal nodes coupled with a rich semantics for propagating values within the tree. Leaves of the tree are annotated with traded items and all nodes are annotated with changes in values (either positive or negative). *TBBL* is designed such that these changes in value are expressed on *trades* rather than the total value of allocations. Examples are provided in Figures 1 and 2.

Consider bid tree T_i from bidder i. Let $\beta \in T_i$ denote a node in the tree, and let $v_i(\beta) \in \mathbb{R}$ denote the value specified at node β (perhaps negative). Let *Leaf*$(T_i) \subseteq T_i$ be the subset of nodes representing the leaves of T_i and let *Child*$(\beta) \subseteq T_i$ denote the children of node β. All nodes except leaves are labeled with the *interval-choose* operator $\mathrm{IC}^y_x(\beta)$. Each leaf β is labeled as a *buy* or *sell*, with units $q_i(\beta, j) \in \mathbb{Z}$ for the good j associated with leaf β, and $q_i(\beta, j') = 0$ otherwise. The same good j may simultaneously occur in multiple leaves of the tree, given the semantics of the tree described below.

The IC operator defines a range on the number of children that can be, and must be, satisfied for node β to be satisfied: an $\mathrm{IC}^y_x(\beta)$ node (where x and y are non-negative integers) indicates that the bidder is willing to pay for the satisfaction of at least x and at most y of his children. With suitable values for x and y the operator can include many logical connectors. For instance: $\mathrm{IC}^n_n(\beta)$ on node β with n children is equivalent to an AND operator; $\mathrm{IC}^n_1(\beta)$ is equivalent to an OR operator; and $\mathrm{IC}^1_1(\beta)$ is equivalent to an XOR operator.[2]

We say that the *satisfaction* of an $\mathrm{IC}^y_x(\beta)$ node is defined by the following two rules:

2. This equivalence implies that *TBBL* can directly express the XOR, OR and XOR/OR languages (Nisan, 2006).

R1 Node β with $\text{IC}_x^y(\beta)$ may be *satisfied* only if at least x and at most y of its children are *satisfied*.

R2 If some node β is *not satisfied*, then none of its children may be *satisfied*.

One can consider **R1** as a "first pass" that defines a set of candidates for satisfaction. This candidate set is then refined by **R2**. Besides defining how value is propagated, by virtue of **R2** our logical operators act as *constraints* on what trades are *acceptable* and provide necessary and sufficient conditions.[3]

Given a tree T_i, the (change in) value of a trade λ is defined as the sum of the values on all satisfied nodes, where the set of satisfied nodes is chosen to provide the *maximal* total value. Let $sat_i(\beta) \in \{0, 1\}$ denote whether node β in tree T_i of bidder i is satisfied, with $sat_i = \{sat_i(\beta), \quad \forall \beta \in T_i\}$. For solution sat_i to be valid for tree T_i and trade λ_i, written $sat_i \in valid(T_i, \lambda_i)$, then rules **R1** and **R2** must hold for all internal nodes $\beta \in \{T_i \backslash Leaf(T_i)\}$ with $\text{IC}_x^y(\beta)$:

$$x \ sat_i(\beta) \leq \sum_{\beta' \in Child(\beta)} sat_i(\beta') \leq y \ sat_i(\beta) \tag{8}$$

Equation (8) enforces the *interval-choose* constraints, by ensuring that no more and no less than the appropriate number of children are *satisfied* for any node that is *satisfied*. The constraint also ensures that any time a node other than the root is satisfied, its parent is also satisfied. We further require, for $sat_i \in valid(T_i, \lambda_i)$, that the total increase in quantity of each item across all satisfied leaves is no greater than the total number of units awarded in the trade:

$$\sum_{\beta \in Leaf(T_i)} q_i(\beta, j) sat_i(\beta) \leq \lambda_{ij}, \qquad \forall j \in G \tag{9}$$

By free disposal, we allow here for a trade to assign additional units of an item over-and-above that required in order to activate leaves in the bid tree. This works for sellers as well as buyers: for sellers a trade is negative and this requires that the total number of items indicated sold in the tree is at least the total number of items "traded away" from the bidder in the trade.

Given these constraints, the total value of trade λ_i, given bid-tree T_i from bidder i, is defined as the solution to an optimization problem:

$$v_i(T_i, \lambda_i) = \max_{sat_i} \sum_{\beta \in T_i} v_i(\beta) sat_i(\beta) \tag{10}$$
$$\text{s.t.} \quad (8), \ (9)$$

Example 1 *Consider an airline operating out of a slot-controlled airport that already owns several morning landing slots, but none in the evening. In order to expand its business the airline wishes to acquire at least two and possibly three of the evening slots. However, it needs to offset the cost of this purchase by selling one of its morning slots. Figure 1 shows a* TBBL *valuation tree for expressing this kind of swap.*

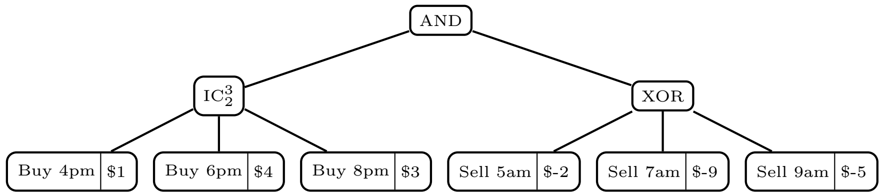


Figure 1: A simple *TBBL* tree for an airline interested in trading landing slots.

In working through numerous examples we frequently found it very cumbersome to capture even simple trades in languages that specified values on allocations, as is the case with all existing languages. Indeed, in earlier work, we demonstrate natural instances for which *TBBL* is *exponentially more concise* than OR* and $\mathcal{L}_{GB}$ (Cavallo et al. 2005). In fact, *TBBL*'s conciseness is incomparable to OR* and $\mathcal{L}_{GB}$ but can be extended in simple ways to strictly dominate these earlier languages.

Winner Determination. The problem of determining an efficient trade given bids is called the *winner determination* (WD) problem. The WD problem in CAs (and thus also in CEs) is NP-hard (Rothkopf et al., 1998). The approach we adopt here is to formulate the problem as a mixed-integer program (MIP), and solve with branch-and-cut algorithms (Nemhauser & Wolsey, 1999). A similar approach has proved successful for solving the WD problem in CAs (de Vries & Vohra, 2003; Boutilier, 2002; Sandholm, 2006).

Given some tree T_i, it is useful to adopt notation $\beta \in \lambda_i$ to denote a node $\beta \in T_i$ that is satisfied by trade λ_i. We can now formulate the WD problem for bid trees $T = (T_1, \ldots, T_n)$ and initial allocation x^0:

$$\begin{aligned} \mathrm{WD}(T, x^0): \ & \max_{\lambda, sat} \sum_i \sum_{\beta \in T_i} v_i(\beta) sat_i(\beta) \\ \text{s.t.} \quad & (2), (3) \\ & sat_i \in valid(T_i, \lambda_i), \qquad \forall i \\ & sat_i(\beta) \in \{0, 1\}, \lambda_{ij} \in \mathbb{Z}, \end{aligned}$$

where $sat = (sat_1, \ldots, sat_n)$. The tree structure is made explicit in this MIP formulation: we have decision variables to represent the satisfaction of nodes and capture the logic of the *TBBL* language through linear constraints; a related approach approach has been considered in application to $\mathcal{L}_{GB}$ (Boutilier, 2002). By doing this, there are $O(nB + mn)$ variables and constraints, where B is the maximal number of nodes in any bid tree. The formulation determines the trade λ while simultaneously determining the value to all bidders by activating nodes in the bid trees.

Payments. Given reported valuation functions $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_n)$ from each bidder, the Vickrey-Clarke-Groves (VCG) (e.g. Krishna, 2002) mechanism collects the following pay-

3. **R1** naturally generalizes the approach taken in $\mathcal{L}_{GB}$, where an internal node is satisfied according to its operator and the subset of its children that are satisfied. The semantics of $\mathcal{L}_{GB}$, however, treat logical operators only as a way of specifying when "added value" (positive or negative) results from attaining combinations of goods. Our use of **R2** also imposes constraints on acceptable trades.

ments from each bidder:

$$p_{\text{vcg},i} = \hat{v}_i(\lambda_i^*) - (V(\hat{v}) - V_{-i}(\hat{v})), \tag{11}$$

where λ^* is the efficient trade, $V(\hat{v})$ is the reported value of this trade and $V_{-i}(\hat{v})$ is the reported value of the efficient trade in the economy without bidder i, where $v_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$. Let us refer to $\Delta_{\text{vcg},i} = V(\hat{v}) - V_{-i}(\hat{v})$ as the *VCG discount*. The problem with the VCG mechanism in the context of a CE is that it may run at a budget deficit with the total payments negative. An alternative payment method is provided by the Threshold rule (Parkes et al., 2001):

$$p_{\text{thresh},i} = \hat{v}_i(\lambda_i^*) - \Delta_{\text{thresh},i}, \tag{12}$$

where the discounts $\Delta_{\text{thresh},i}$ are picked to minimize $\max_i(\Delta_{\text{vcg},i} - \Delta_{\text{thresh},i})$ subject to $\Delta_{\text{thresh},i} \leq \Delta_{\text{vcg},i}$ for all i and $\sum_i \Delta_{\text{thresh},i} \leq V(\hat{v})$. Threshold payments are exactly budget balanced and minimize the maximal deviation from the VCG outcome across all balanced rules.

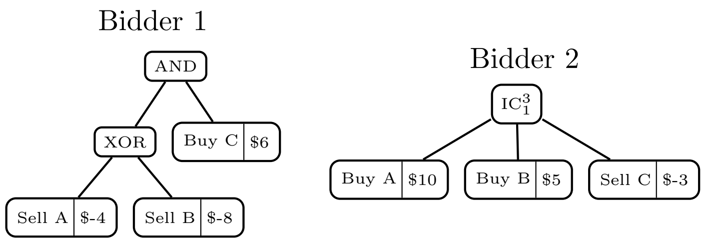


Figure 2: Two bidders and three items $\{A, B, C\}$. The efficient trade is for bidder 1 to sell A and buy C.

Example 2 *Consider the two bidders in Figure 2. Bidder 1 will potentially sell one of his items (A or B) if he can get Bidder 2's item, C, at the right price. Bidder 2 is interested in buying one or both of Bidder 1's items and also selling his own item. We consider each of the possible trades: If Bidder 1 trades A for C he gets \$2 of value and Bidder 2 gets \$7. If Bidder 1 trades B for C he gets \$-2 of value and Bidder 2 gets \$2. And if no trade occurs both bidders get \$0 value. Therefore the efficient trade is to swap A for C.*

Because the efficient trade creates a surplus of \$9 and removing either bidder results in the null trade, both bidders have a Vickrey discount of \$9. Thus if we use VCG payments, Bidder 1 pays \$2-\$9=\$-7 and Bidder 2 pays \$7-\$9=\$-2 and the exchange runs at a deficit. The Threshold payment rule chooses payments that minimally deviate from VCG while maintaining budget balance. This minimization reduces the discounts to \$4.50, and thus Bidder 1 pays \$2-\$4.50=\$-2.50 and Bidder 2 pays \$7-\$4.50=\$2.50.

4. Step Two: Making the Exchange Iterative

Having defined a sealed-bid, *TBBL*-based exchange we can now modify the design to make it iterative. Rather than provide an exact valuation for all interesting trades, a bidder

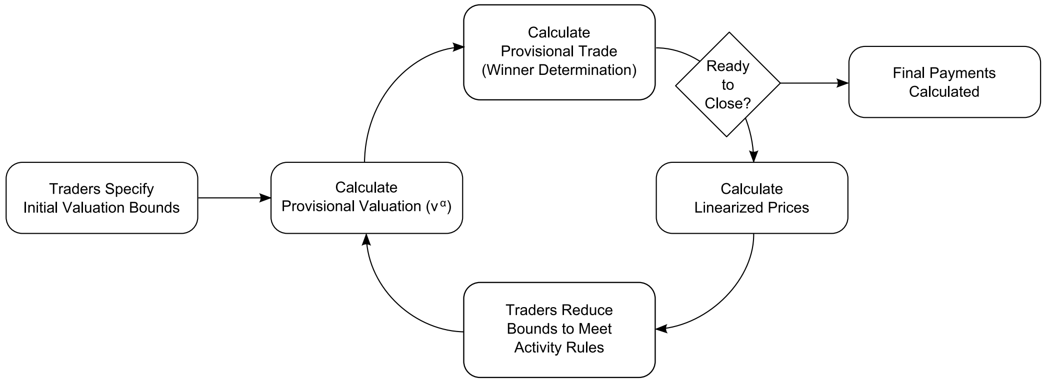


Figure 3: ICE system overview

annotates a single *TBBL* tree with upper and lower bounds on his valuation. The ICE mechanism then proceeds in rounds, as illustrated in Figure 3.

ICE is a *proxied design* in which each bidder has a proxy to facilitate his valuation refinement. In each round, a bidder responds to prices in interacting with his proxy agent in order to tighten the bounds on his *TBBL* tree and meet the activity rules. The exchange chooses a *provisional valuation profile* (denoted $v^\alpha = (v_1^\alpha, \ldots, v_n^\alpha)$ in the figure), with the valuation v_i^α for each bidder picked to fall within the bidder's current valuation bounds (and to tend towards the lower valuation bound as progress is made towards determining the final trade). Then, the exchange computes a *provisional trade* λ^α and checks whether the conditions for moving to a last-and-final round are satisfied. Approximate equilibrium prices are then computed based on valuation profile v^α and trade λ^α and a new round begins. In the last-and-final round, the final payments and the trade are computed in terms of *lower* valuations; the semantics are that these lower bounds guarantee that a bidder is willing to pay at least this amount (or receive a payment of this amount) in order to complete the trade.

Let $\underline{v}_i$ and $\overline{v}_i$ denote the lower and upper valuation functions reported by bidder i in a particular round of ICE, and adopt $\text{WD}(v)$ to denote the WD problem for valuation profile $v = (v_1, \ldots, v_n)$. ICE is parameterized by a target approximation error $\Delta^* \in (0, 1]$, which requires that the total value from the optimal trade $\underline{\lambda}$ given the current lower-bound valuation profile (i.e., $\underline{\lambda}$ solves $\text{WD}(\underline{v})$) is close to the total value from the efficient trade λ_i^*:

$$\text{EFF}(\underline{\lambda}) = \frac{\sum_i v_i(\underline{\lambda}_i)}{\sum_i v_i(\lambda_i^*)} = \frac{v(\underline{\lambda})}{v(\lambda^*)} \geq \Delta^* \tag{13}$$

However, the true valuation v and thus the trade λ^* are uncertain within ICE and thus we will later introduce techniques to estblish this bound.

In each round, ICE goes through the following steps:

1. If this is the last-and-final round, then implement the trade that solves $\text{WD}(\underline{v})$ and collect Threshold payments defined on valuations $\underline{v}$. STOP.

ELSE,

2. Solve WD($\underline{v}$) to obtain $\underline{\lambda}$. Use valuation bounds and prices to determine a lower-bound, ω^{eff}, on the allocative efficiency EFF($\underline{\lambda}$) of $\underline{\lambda}$. If $\omega^{\text{eff}} \geq \Delta^*$ then the next round will be designated the last-and-final round.

3. Set $\alpha \in [0, 1]$, with α tending to 1 as ω^{eff} tends to 1, and provisional valuation profile $v^\alpha = (v_1^\alpha, \ldots, v_n^\alpha)$, where $v_i^\alpha(\lambda_i) = \alpha \underline{v}_i(\lambda_i) + (1-\alpha)\overline{v}_i(\lambda_i)$, expressed with a *TBBL* tree in which the value on node $\beta \in T_i$ is $v_i^\alpha(\beta) = \alpha \underline{v}_i(\beta) + (1-\alpha)\overline{v}_i(\beta)$.

4. Solve WD(v^α) to find *provisional trade* λ^α, and determine Threshold payments for provisional valuation profile, v^α.

5. Compute linear prices, $\pi \in \mathbb{R}^m_{\geq 0}$, that are approximate CE prices given valuations v^α and trade λ^α, breaking ties to best approximate the provisional Threshold payments and finally to minimize the difference in price between items.

6. Report (λ_i^α, π) to each bidder $i \in N$, and whether or not the next round is last-and-final.

In transitioning to the next round, the proxy agents are responsible for guiding bidders to make refinements to their lower- and upper-bound valuations in order to meet activity rules that ensure progress towards the efficient trade across rounds. In what follows, we (a) extend *TBBL* to capture lower and upper valuation bounds, (b) describe our two activity rules, (c) explain how we compute price feedback, (d) provide our main theoretical results. In developing theoretical and experimental results about ICE we assume *straightforward* bidders, so that bidders refine upper and lower bounds on valuations to keep their true valuation consistent with the bounds.

Extending *TBBL*. We first extend *TBBL* to allow bidder i to report a lower and upper bound $(\underline{v}_i(\beta), \overline{v}_i(\beta))$ on the value of each node $\beta \in T_i$, which in turn induces valuation functions $\underline{v}_i(T_i, \lambda_i)$ and $\overline{v}_i(T_i, \lambda_i)$, using the exact same semantics as in (10). The bounds on a trade can be interpreted as bounding the payment that the bidder considers acceptable. The bidder commits to complete the trade for a payment less than or equal to the lower-bound and to refuse to complete a trade for any payment greater than the upper-bound. The exact value, and thus true willingness-to-pay, remains unknown except when $\underline{v}_i(\beta) = \overline{v}_i(\beta)$ on all nodes. We say that bid-tree T_i for bidder i is *well-formed* if $\underline{v}_i(\beta) \leq \overline{v}_i(\beta)$ for all nodes $\beta \in T_i$. In this case we also have $\underline{v}_i(T_i, \lambda_i) \leq \overline{v}_i(T_i, \lambda_i)$ for all trades λ_i. We refer to the difference $\overline{v}_i(\beta) - \underline{v}_i(\beta)$ as the *value uncertainty* on node β. The efficient trade can often be determined with only partial information about bidder valuations. Consider the following simple variant on Example 2:

Example 3 *The structure of the bidders' trees in Figure 4 is the same as in Example 2 but the nodes are annotated with bounds. Let $x \in [3, 8]$ denote Bidder 1's true value for "buy C" and $y \in [-4, -1]$ denote Bidder 2's true value for "sell C." The three feasible trades are: (1) trade A and C, (2) trade B and C, (3) no trade. The first trade is already provably efficient. Fixing x and y, its minimal value is $-4+9+x-y$ and this is at least $-5+7+x-y$, the value of the second trade. Moreover, its worst-case value is $-4+9+3-4 \geq 0$, the value of the null trade.*

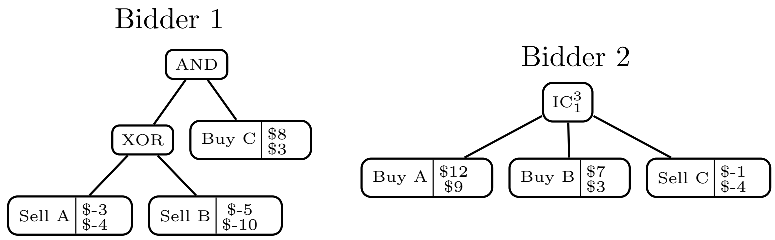


Figure 4: Two bidders, each with partial value information defined on their bid tree. One can already prove that the efficient trade is for bidder 1 to sell A and buy C.

4.1 Activity Rules

Activity rules are used to guide the preference elicitation process in each round of ICE. Without an activity rule, a rational bidder would likely wait until the very last moment to revise his valuation information, free-riding on the price discovery enabled by the bids of other participants. If every bidder were to behave this way then the exchange would reduce to a sealed-bid mechanism and lose its desirable properties.[4] Thus, activity rules are critical in mitigating opportunities for strategic behavior.[5]

ICE employs two activity rules. In presenting our activity rules, we will not specify the explicit consequences of failing to meet an activity rule. One simple possibility is that the default action is to automatically set the upper valuation bound on every node in a bid tree to the maximum of the "provisional price on a node"[6] and the lower-bound value on that node. This is entirely analogous to when a bidder in an ascending-clock auction stops bidding at a price: he is not permitted to bid at a higher price again in future rounds.

Modified Revealed-Preference Activity Rule (MRPAR). The first rule, MRPAR, is based on a simple idea. We require bidders to refine their valuation bounds in each round, so that there is some trade that is optimal (i.e., maximizes surplus) for the bidder given the current prices and for all possible valuations consistent with the bounds. MRPAR is loosely based around the revealed-preference based activity rule, advocated for the clock-proxy auction in a one-sided CA (Ausubel et al., 2006).

Let $v'_i \in T_i$ for *TBBL* tree T_i denote that valuation v'_i is consistent with the value bounds in the tree. If the bounds are tight everywhere, then v'_i is exactly the valuation function defined by tree T_i. A simple variant (RPAR), requires that there is enough information in valuation bounds to establish that one trade is weakly preferred to all other trades at the prices, i.e.

$$\exists \breve{\lambda}_i \in \mathcal{F}_i(x^0) \text{ s.t. } v'_i(\breve{\lambda}_i) - p^\pi(\breve{\lambda}_i) \geq v'_i(\lambda'_i) - p^\pi(\lambda'_i), \quad \forall v'_i \in T_i, \forall \lambda'_i \in \mathcal{F}_i(x^0) \qquad \text{(RPAR)}$$

Note that a bidder can always meet this rule by defining an *exact* valuation $\hat{v}_i$ and tight value bounds on every node in his bid tree; in this case, trade $\breve{\lambda}_i \in \arg\max_{\lambda_i \in \mathcal{F}_i(x^0)}[\hat{v}_i(\lambda_i) -$

4. This problem has been evocatively described as the "snake in the grass" problem. See Kwerel's forward in Milgrom's book (2004).
5. There is no conflict here with our assumption about straightforward bidding: we design for the strategic case despite assuming straightforward bidding to provide for tractable theoretical and experimental analysis; moreover, the presence of activity rules helps to motivate straightforward bidding.
6. The provisional price on a node is defined as the minimal total price across all feasible trades for which the subtree rooted at the node is satisfied.

$p^\pi(\lambda_i)]$ satisfies RPAR. We say that prices π are *strict EQ prices* for $(v^\alpha, \lambda^\alpha)$ when:

$$v_i^\alpha(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha) > v_i^\alpha(\lambda_i') - p^\pi(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0) \setminus \{\lambda_i^\alpha\}\,, \tag{14}$$

for every bidder $i \in N$.

Theorem 3 *If prices π are strict EQ prices for provisional valuation profile v^α and trade λ^α, and every bidder i retains v_i^α in his bid tree after meeting RPAR, then trade λ^α is efficient when all bidders are straightforward.*

Proof: Fix bidder i. Let $\breve{\lambda}_i$ denote the trade that satisfies RPAR. Because v_i^α is consistent with the revised bid tree of bidder i, we have:

$$v_i^\alpha(\breve{\lambda}_i) - p^\pi(\breve{\lambda}_i) \geq v_i^\alpha(\lambda_i') - p^\pi(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0). \tag{15}$$

Moreover, we must have $\breve{\lambda}_i = \lambda_i^\alpha$, because $v_i^\alpha(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha) > v_i^\alpha(\lambda_i') - p^\pi(\lambda_i')$ by the strictness of prices. Instantiating RPAR with this trade, and with true valuations $v_i \in T_i$ (since bidders are straightforward), we have:

$$v_i(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha) \geq v_i(\lambda_i') - p^\pi(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0), \tag{16}$$

from which prices p^π are EQ prices with respect to true valuations. The efficiency claim then follows from the welfare theorem, Theorem 1. □

In particular, the provisional trade is efficient given strict EQ prices when every bidder meets the rule without modifying his bounds in any way. Strict EQ prices are required to prevent problems involving ties:

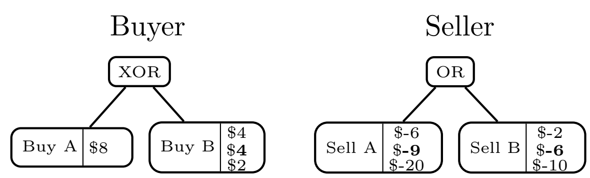


Figure 5: An Example to illustrate the failure of the simple RPAR rule without strict EQ prices. True values are shown in bold and are such that the efficient outcome is no trade.

Example 4 *The* TBBL *trees shown in Figure 5 will have no trade occur at the truthful valuation (which is indicated in bold between the value bounds). However, suppose $\alpha = 0$ so that at the provisional valuations it is efficient for A to be traded. Prices $\pi = (6, 2)$ are EQ (but not strict EQ) prices given v^α and λ^α, with the buyer indifferent between buying A and buying B and the seller indifferent between selling A, selling A and B, or making no sale. The buyer passes RPAR without changing his bounds because the bounds already establish that he (weakly) prefers A to B, and prefers A to no trade, at all possible valuations. Similarly, the seller passes RPAR without changing his bounds because the bounds establish that he weakly prefers no trade to selling any combination of A and B given the current prices. Thus, we have no activity even though the current provisional trade is inefficient.*

In order to better handle these sorts of ties, we slightly strengthen RPAR to *modified RPAR* (MRPAR), which requires that there exists some $\breve{\lambda}_i \in \mathcal{F}_i(x^0)$ such that

$$\theta_i^\pi(\breve{\lambda}_i, \lambda_i', v_i') \geq 0, \quad \forall v_i' \in T_i, \forall \lambda_i' \in \mathcal{F}_i(x^0) \tag{17}$$

$$\textbf{and} \text{ either } \breve{\lambda}_i = \lambda_i^\alpha \quad \textit{or} \quad \theta_i^\pi(\breve{\lambda}_i, \lambda_i^\alpha, v_i') > 0,\ \forall v_i' \in T_i. \tag{18}$$

where $\theta_i^\pi(\lambda_i, \lambda_i', v_i') = v_i'(\lambda_i) - p^\pi(\lambda_i) - (v_i'(\lambda_i') - p^\pi(\lambda_i'))$ denotes the profit to bidder i for trade λ_i over λ_i' given v_i' and prices π. (17) is RPAR and the additional requirements enforce that the satisfying trade $\breve{\lambda}_i$ is either λ_i^α or strictly preferred to λ_i^α. This need to show a strict preference over λ_i^α prevents the deadlock shown in Example 4. The seller has shown only a *weak* preference for not trading over selling A. With MRPAR, the seller must also show that he strictly prefers $\breve{\lambda}_i$, in this case by reducing the upper-bounds on both A and B, thus ensuring progress.

The actual rule adopted in ICE is δ-MRPAR, parameterized with *accuracy parameter* $\delta \geq 0$, and providing a relaxation of MRPAR which is useful even when there are no exact EQ prices defined with respect to $(\lambda^\alpha, v^\alpha)$ in some round.

Definition 4 *Given provisional trade λ^α, linear prices π, and accuracy parameter $\delta \geq 0$, δ-MRPAR requires that every bidder i refines his value bounds so that his* TBBL *tree T_i satisfies:*

$$\theta_i^\pi(\lambda_i^\alpha, \lambda_i', v_i') \geq -\delta, \quad \forall v_i' \in T_i, \forall \lambda_i' \in \mathcal{F}_i \tag{19}$$

or*, that there is some $\breve{\lambda}_i \in \mathcal{F}_i(x^0)$ such that*

$$\theta_i^\pi(\breve{\lambda}_i, \lambda_i', v_i') \geq 0, \quad \forall v_i' \in T_i, \forall \lambda_i' \in \mathcal{F}_i(x^0) \tag{20}$$

$$\theta_i^\pi(\breve{\lambda}_i, \lambda_i^\alpha, v_i') > \delta, \quad \forall v_i' \in T_i \tag{21}$$

It is a simple matter to check that δ-MRPAR reduces to MRPAR for $\delta = 0$. Phrasing the description to allow for the rule to be interpreted with and without the δ relaxation, δ-MRAPR requires that *each bidder must adjust his valuation bounds to establish that the provisional trade is [within δ of] maximizing profit for all possible valuations* (19)*, or some other trade satisfies RPAR* (20) *and is strictly preferred [by at least δ] to the provisional trade* (21). Just as for RPAR, one can show that a bidder can always meet δ-MRPAR (for any δ) by defining an exact valuation.[7]

Lemma 1 *If every bidder i meets δ-MRPAR without precluding v_i^α from his updated bid tree, and prices π are δ-approximate EQ prices with respect to provisional valuation profile v^α and trade λ^α, and bidders are straightforward, then the provisional trade is a $2\min(M, \frac{n}{2})\delta$-approximate efficient trade.*

7. Let v_i denote this valuation. If δ-MRPAR is not satisfied via (19) then $\breve{\lambda}_i \in \arg\max_{\lambda_i \in \mathcal{F}_i(x^0)}[v_i(\lambda_i) - p^\pi(\lambda)]$ will satisfy δ-MRPAR. This satisfies (20) by construction. Now, let λ_i' denote the trade with $v_i(\lambda_i') - p^\pi(\lambda_i') > v_i(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha) + \delta$. We have $v_i(\breve{\lambda}_i) - p^\pi(\breve{\lambda}_i) \geq v_i(\lambda_i') - p^\pi(\lambda_i') > v_i(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha) + \delta$, and (21).

Proof: Fix bidder i. By δ-EQ, we have $\theta_i^\pi(\lambda_i^\alpha, \lambda_i', v_i^\alpha) \geq -\delta$ for all $\lambda_i' \in \mathcal{F}_i(x^0)$. Consider any $\breve{\lambda}_i \neq \lambda_i^\alpha$. Because v_i^α remains in the bid tree, we must have $\theta_i^\pi(\breve{\lambda}_i, \lambda_i^\alpha, v_i^\alpha) \leq \delta$ and δ-MRPAR cannot be satisfied via (20) and (21). Therefore, δ-MRPAR is satisfied for every bidder via (19) and with provisional trade λ^α the satisfying trade. Therefore we prove that prices, π, are δ-approximate EQ prices for all valuations, and including the true valuation since bidders are straightforward and this is within their bounds. The efficiency of the trade follows from Theorem 2. □

This in turn provides a simple proof for the efficiency of ICE when approximate CE prices exist upon termination. Suppose that ICE is defined to terminate as soon as prices are δ-accurate and v^α is retained in the bid tree by all bidders in meeting the activity rule, or when quiescence is reached and no bidder refines his bounds in meeting the rule. In this variation, the provisional trade λ^α is the trade finally implemented.

Theorem 4 *ICE with δ-MRPAR is $2\min(M, \frac{n}{2})\delta$-efficient when prices are δ-accurate with respect to $(v^\alpha, \lambda^\alpha)$ upon termination and bidders are straightforward.*

Proof: When ICE terminates either (a) prices are δ-accurate and v^α is retained in the bid tree by all bidders and we can appeal directly to Lemma 1, or (b) no bidder refines his bounds in meeting δ-MRPAR, in which case v_i^α remains in the space of valuations consistent with the bid tree for each bidder. □

We also have the following simple corollary, which considers the property of ICE for a domain in which approximately accurate EQ prices exist:

Corollary 1 *ICE with δ-MRPAR is $2\min(M, \frac{n}{2})\delta$-efficient when δ-accurate competitive equilibrium prices exist for all valuations in the valuation domain and when all bidders are straightforward.*

Specializing to domains in which exact EQ prices exist (e.g., for unit-demand preferences as in the assignment model of Shapley and Shubik, 1972; see also the work of Bikhchandani and Mamer, 1997) then ICE with MRPAR is efficient for straightforward bidders.

(a) Passes δ-MRPAR (b) Fails δ-MRPAR

Figure 6: δ-MRPAR where the provisional trade is "Buy A", $\pi_A = 3, \pi_B = 4$ and $\delta = 2$

Example 5 *To illustrate the δ-MRPAR rule consider a single bidder with a valuation tree as in Figure 6(a). Suppose the provisional trade λ_i^α allocates A to the bidder, and with prices $\pi_A = 3, \pi_B = 4$ and $\delta = 2$. Here the bidder has satisfied δ-MRPAR because the guaranteed \$2-\$3=\$-1 payoff from A is within δ of the possible \$5-\$4=\$1 payoff from B. Now consider Figure 6(b), with a relaxed upper-bound on "buy B" of \$8. Now the bidder fails δ-MRPAR because the guaranteed \$-1 payoff from A is not within δ of the possible payoff from B of \$8-\$4=\$4. Let $[x, v]$ and $[w, y]$ denote the lower and upper bounds, on "buy A" and "buy B" respectively, as revised in meeting the rule. To pass the rule, the bidder has two choices:*

- *Demonstrate λ_i^α is the best response. To do so the bidder will need to adjust x and y to make $x - 3 \geq y - 4 - 2 \Rightarrow y - x \leq 3$; e.g., values $x = \$2, y = \$5$ solve this, as in Figure 6(a), as do many other possibilities.*

- OR *Demonstrate that another trade (e.g., "buy B") is more than \$2 better than λ_i^α, i.e., $w - 4 > v - 3 + 2 \Rightarrow w - v > 3$, and "buy B" is weakly better than the null trade, i.e., $w - 4 \geq 0$. For instance, if the bidder's true values are $v_A = \$3, v_B = \$8$ then $x \leq 3 \leq v$ and $w \leq 8 \leq y$ and the rule cannot be satisfied in the first case. But, the buyer can establish that "buy B" is his best-response, e.g., by setting $v = \$4, w = \$7$, or $v = \$3, w = \$6$.*

Remark: Computation and Bidder Feedback. The definition of MRPAR naively suggests that checking for compliance requires explicitly considering all valuations $v_i' \in T_i$ and all trades $\lambda_i' \in \mathcal{F}_i(x^0)$. Fortunately, this is not necessary. We present in the Appendix a method to check MRPAR given prices π, provisional trade λ_i^α and bid tree T_i by solving three MIPs. Moreover, we explain that the solution to these MIPs also provides nice feedback for bidders. ICE can automatically identify a set of nodes at which a bidder needs to increase his lower bound and a set of nodes at which a bidder needs to decrease his upper bound in meeting MRPAR.

Delta Improvement Activity Rule (DIAR). With only δ-MRPAR, it is quite possible for ICE to get stuck, with all bidders satisfying the activity rule without changing their bounds, but with the prices less than δ accurate (with respect to $(\lambda^\alpha, v^\alpha)$). Therefore, we need an activity rule that will continue to drive a reduction in value uncertainty, i.e., the gap between upper bound values and lower bound values, even in the face of inaccurate prices, and ideally in a way that remains "price-directed" in the sense of using prices to determine which trades (and in turn which nodes in *TBBL* trees) each bidder should be focused on.

We introduce for this purpose a second (and novel) activity rule (DIAR), which fills this role by requiring bidders to reveal information so as to improve price accuracy and, in the limit, full information on the nodes that matter. Defined this way, the DIAR rule very nicely complements the δ-MRPAR rule. Because we can establish the efficiency of the provisional trade directly via the valuation bounds, as we will see in Section 4.3, we do not actually need fully accurate prices in order to close the exchange. Thus, the DIAR rule does not imply that bidders will reveal full information. Rather, the presence of DIAR ensures both good performance in practice as well as good theoretical properties. In our experiments we enable DIAR in all rounds of ICE, and it fires in parallel with δ-MRPAR. In practice, we see that most of the progress in refining valuation information occurs due to δ-MRPAR, and that *all* the progress in early rounds occurs due to δ-MRPAR. Experimental support for this is provided in Section 6.[8]

Before providing the specifics of DIAR, we identify a node $\beta \in T_i$ in the bid tree of bidder i as *interesting* for some fixed instance (v, x^0), when the node is satisfied in *some* feasible trade. We have the following simple lemma:

8. In a variation on the way ICE is defined, DIAR could be used only in rounds in which the price error for the provisional valuation and trade is greater than the error associated with δ-MRPAR. This is because δ-MRPAR is sufficient for approximate efficiency when prices are accurate enough.

Lemma 2 *If there is no value uncertainty on any interesting nodes in the bid trees of any biders, and bidders are straightforward, then λ^α is efficient.*

Proof: No value uncertainty and thus exact information about the value on all *interesting* nodes implies that the difference in value is exactly known between all pairs of feasible trades because for all uninteresting nodes, either the node is never satisfied in any trade (and thus its value does not matter) or the node is satisfied in every trade and thus its actual value does not matter in defining the difference in value between pairs of trades. Only the difference in value between pairs of trades is important in determining the efficient trade. $\square$

DIAR focuses a bidder in particular on interesting nodes that correspond to trades for which the pricing error is large, and where this error could still be reduced by refining the valuation bounds on the node. Given prices π and provisional trade λ_i^α, the main focus of DIAR is the following upper-bound $\overline{\delta}_i^k$, on the amount by which prices π might misprice some trade $\lambda_i^k \in \mathcal{F}_i(x^0)$ with respect to bidder i's true valuation:

$$\overline{\delta}_i^k = \max_{v_i' \in T_i} [v_i'(\lambda_i^k) - p^\pi(\lambda_i^k) - (v_i'(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))] \tag{22}$$

We call this the "DIAR error on trade λ_i^k", and note that it depends on the current prices as well as the current bid tree and provisional trade, but not the true valuation which is unknown to the center. The DIAR error provides an upper bound on the additional payoff that the bidder could achieve from trade λ_i^k over trade λ_i^α. If we order trades, $\lambda_i^1, \lambda_i^2, \ldots$, so that λ_i^1 has maximal DIAR error, then $\overline{\delta}_i^1 \geq \delta_i$, where $\delta_i = \max_{\lambda_i' \in \mathcal{F}_i(x^0)} [v_i^\alpha(\lambda_i') - p^\pi(\lambda_i') - (v_i^\alpha(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))]$ is the pricing error with respect to the provisional trade and provisional valuation profile. This is the error that the pricing algorithm is designed to minimize in each round, and the same error that is used in Theorem 2 in reference to δ-accurate prices. Thus, we see that *the maximal DIAR error also bounds the amount by which prices are approximate EQ prices*, and that if $\overline{\delta}_i^1 \leq 0$ for all bidders i then the current prices π are exact EQ prices with respect to $(\lambda^\alpha, v^\alpha)$.

To satisfy DIAR a bidder must reduce the DIAR error on the trade with the largest error for which the error can be reduced (some error may just be intrinsic given the current prices and not because of uncertainty about the bidder's valuation), or establish by providing exact value information throughout the tree that none of the DIAR error on any trades is due to value uncertainty. Figure 7 illustrates the difference between MRPAR and DIAR. A bidder can satisfy MRPAR by making it clear that the lower bound on payoff from some trade is greater than the upper bound on all other trades, but still leave large uncertainty about value. DIAR requires that a bidder also refine this upper bound if it is on a node that corresponds to a trade for which the DIAR error (and thus potentially the actual approximation in prices) is large. The rule is illustrated in Figure 8.

DIAR is parameterized by some $\epsilon \geq 0$. We refer to the formal rule as ϵ-DIAR:

Definition 5 *To satisfy ϵ-DIAR given provisional trade λ_i^α and prices π, the bidder must modify his valuation bounds to:*
(a) reduce the DIAR error on some trade, $\lambda_i^j \in \mathcal{F}_i(x^0)$, by at least ϵ **and**
*(b) prove that error $\overline{\delta}_i^k$ **cannot** be improved by ϵ for all trades $\lambda_i^k \in \mathcal{F}_i(x^0)$ for $1 \leq k < j$,*
or *(c) establish that $\overline{\delta}_i^k$ cannot be improved by ϵ on **any** trade $\lambda_i^k \in \mathcal{F}_i(x^0)$.*

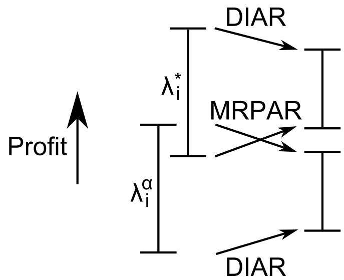


Figure 7: Stylized effect of MRPAR and DIAR on the bounds of the λ_i^α and λ_i^* trades

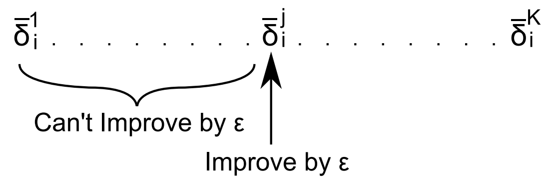
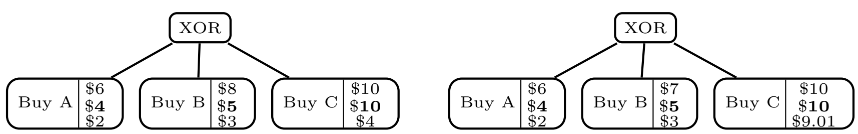


Figure 8: Trades for bidder i, ordered with DIAR error reducing from left to right. The bidder must reduce, by at least ϵ, the DIAR error on the trade with the greatest error for which this is possible and prove (via valuation bounds) that it is impossible to improve by ϵ any trades with larger error.

In particular, even if the bidder is in case (c) above, he will still be forced to narrow his bounds and progress will be made towards bounding efficiency. In practice, we define the ϵ parameter to be large at the start and smaller in later rounds.



(a) Fails DIAR (b) Passes DIAR

Figure 9: Respecting DIAR where the provisional trade is "Buy A", $\pi_A = 4, \pi_B = 5, \pi_C = 6$ and $\epsilon = 1$

Example 6 *Consider the tree in Figure 9(a) when the provisional trade is "buy A", prices* $\pi = (\$4, \$5, \$6)$ *and DIAR parameter* $\epsilon = 1$. *The DIAR error on each trade, defined via* (22), *and listed in decreasing order, are:*

$$
\begin{aligned}
C &\rightarrow \bar{\delta}^1 = (\$10 - \$6) - (-\$2) = \$6 \\
B &\rightarrow \bar{\delta}^2 = (\$8 - \$5) - (-\$2) = \$5 \\
\emptyset &\rightarrow \bar{\delta}^3 = (\$0 - \$0) - (-\$2) = \$2 \\
A &\rightarrow \bar{\delta}^4 = (\$2 - \$4) - (-\$2) = \$0,
\end{aligned}
$$

where $-\$2 = \$2 - \$4$ *is the worst-case profit from the provisional trade. Now, we see that* $\bar{\delta}^1$ *cannot be made smaller by lowering the upper-bound on leaf "buy C" because this bound* *is already tight against the truthful value of \$10. Instead the bidder must demonstrate that a decrease of $\epsilon = 1$ is impossible by raising the lower bound on "buy C" to 9.01. However $\overline{\delta}^2$ can be decreased by $\epsilon = 1$, by reducing the upper-bound on "buy B" from 8 to 7, giving us the tree in Figure 9(b).*

Lemma 3 *When ICE incorporates DIAR, a straightforward bidder must eventually reveal complete value information on all interesting nodes in his bid tree as $\epsilon \to 0$.*

Proof: Fix provisional trade λ_i^α and consider trade, $\lambda_i^1 \in \mathcal{F}_i(x^0) \neq \lambda_i^\alpha$, with the maximal DIAR error. Continue to assume straightforward bidders. Recall that $v_i(\beta)$ denotes a bidder's true value on node β in his *TBBL* tree. By case analysis on nodes $\beta \in T_i$, meeting the DIAR rule on this trade as $\epsilon \to 0$ requires:

(i) Nodes $\beta \in \lambda_i^1 \setminus \lambda_i^\alpha$. Decrease the upper-bound to $v_i(\beta)$, the true value, to reduce the error. Increase the lower-bound to $v_i(\beta)$ to prove that further progress is not possible.

(ii) Nodes $\beta \in \lambda_i^\alpha \setminus \lambda_i^1$. Increase the lower-bound to $v_i(\beta)$, the true value, to reduce the error. Decrease the upper-bound to $v_i(\beta)$ to prove that further progress is not possible.

(iii) Nodes $\beta \in \lambda_i^\alpha \cap \lambda_i^1$. No change is required.

(iv) Nodes $\beta \notin \lambda_i^1 \cup \lambda_i^\alpha$. No change is required.

Continue to fix some λ_i^α, and consider now the impact of DIAR as $\epsilon \to 0$ and as the rule is met for successive trades, moving from λ_i^1 to λ_i^2 and onwards. Eventually, the value bounds on all nodes $\beta \notin \lambda_i^\alpha$ but *in at least one other feasible trade* are driven to truth by (i), and the value bounds on all nodes $\beta \in \lambda_i^\alpha$ *but not in at least one other feasible trade* are driven to truth by (ii). Noting that the null trade is always feasible, the bidder will ultimately reveal complete value information except on nodes that are not satisfied in any feasible trade. □

Putting this together we have the following simple theorem, which considers the convergence property of ICE when DIAR is the only activity rule.

Theorem 5 *ICE with the ϵ-DIAR rule will terminate with the efficient trade when all bidders are straightforward and as $\epsilon \to 0$.*

Proof: Immediate by Lemma 2 and Lemma 3. □

In practice, we use both δ-MRPAR and DIAR and the role of DIAR is to ensure convergence in instances for which there do not exist good, supporting EQ prices. The use of DIAR does *not* lead, in any case, to full revelation of bidder valuations because we can prove efficiency directly in terms of valuation bounds on different trades (see Section 4.3).

Remark: Computation and Bidder Feedback. We present in the Appendix a method to check ϵ-DIAR given prices π, provisional trade λ_i^α, the bidder's bid tree from the past round and proposed new bid tree by solving two MIPs. Moreover, the solution to these MIPs also provides nice feedback for bidders. ICE can automatically identify the trade, and in turn the corresponding nodes in the bid tree, for which the bidder must provide more information.

4.2 Generating Linear Prices

Given the provisional trade λ^α, provisional valuations v^α, and given that provisional payments have also been determined (according to the payment rule, such as Threshold, adopted in the exchange), approximate clearing prices are computed in each round according to the following rules:

I: Accuracy (ACC). First, we compute prices that minimize the maximal error in the best-response constraints across all bidders.

II: Fairness (FAIR). Second, we break ties to prefer prices that minimize the maximal deviation from provisional payments across all bidders.

III: Balance (BAL). Third, we break ties to prefer prices that minimize the maximal difference in price across all items.

Taken together, these steps are designed to promote the informativeness of prices in driving progress across rounds. Balance is well motivated in domains where items are more likely to be similar in value than dissimilar, preferring prices to be similar across items and rejecting extremal prices. Note that these prices may ascend or descend from round to round – but that they will in general tend towards increasing accuracy, as we shall see experimentally in Section 6.

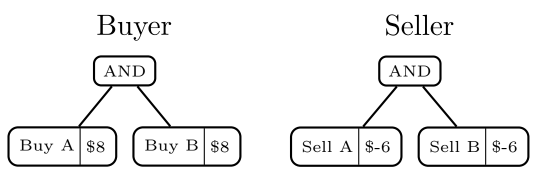


Figure 10: A simple example to illustrate pricing. ACC prices AB between \$12 and \$16, FAIR narrows this to \$14 and BAL requires $A = \$7$, $B = \$7$

Example 7 *Consider the example in Figure 10 with one buyer interested in buying AB and one seller interested in selling AB. Here the buyer's and seller's values for each item are 8 and -6 respectively. The efficient outcome given these values is for the trade to complete. ACC requires* $12 \leq \pi_A + \pi_B \leq 16$, *and thus allows a range of prices. The Threshold payment splits the difference, so that the buyer pays 14 to the seller and so FAIR adds the constraint* $\pi_A + \pi_B = 14$. *Finally, BAL requires* $\pi_A = \pi_B = 7$.

Each of the three stages occur in turn. In the interest of space, here we only present the basic formulation of the Accuracy stage: We define maximally accurate EQ prices by first considering the following LP:

$$\begin{aligned} \delta^*_{\text{acc}} = \min_{\pi, \delta_{\text{acc}}} \ & \delta_{\text{acc}} \\ \text{s.t.} \quad v_i^\alpha(\lambda'_i) - \sum_j \pi_j \lambda'_{ij} \leq v_i^\alpha(\lambda_i^\alpha) - \sum_j \pi_j \lambda_{ij}^\alpha + \delta_{\text{acc}}, & \quad \forall i, \forall \lambda'_i \in \mathcal{F}_i(x^0) \\ \delta_{\text{acc}} \geq 0, & \\ \pi_j \geq 0, \quad \forall j \in G & \end{aligned} \tag{23}$$

These prices minimize the maximal loss in payoff across all bidders for trade λ^α compared to the trade that a bidder would most prefer given provisional valuation v^α, i.e., minimize the maximal value of $\theta_i^\pi(\lambda_i^*, \lambda_i^\alpha, v_i^\alpha)$, where $\lambda_i^* = \arg\max_{\lambda_i \in \mathcal{F}_i(x^0)}[v_i^\alpha(\lambda_i) - p^\pi(\lambda_i)]$. Prices that solve this LP are then refined lexicographically, fixing the worst-case pricing error (ACC) and then working down to try to additionally minimize the next largest pricing error and so on. Given maximally accurate prices, this then triggers a series of lexicographical refinements to best approximate the payments (FAIR) without reducing the pricing accuracy, and eventually a series of lexicographical refinements to try to maximally balance prices across distinct items (BAL). In addition to further improving the quality of the prices, this process also ensures uniqueness of prices.

Each of the Accuracy, Fairness and Balance problems have an exponential number of constraints because the price accuracy constraints (23) (which are carried forward into the subsequent stages) are defined over all trades $\lambda_i' \in \mathcal{F}_i(x^0)$ and all bidders i. It is therefore infeasible to even write these problems down. Rather than solve them explicitly, we use *constraint generation* (e.g. Bertsimas & Tsitsiklis, 1997) and dynamically generate a sufficient subset of constraints. Constraint generation (CG) considers a relaxed program that only contains a manageable subset of the constraints, and solves this to optimality. Given a solution to this relaxed program, a subproblem is used to either prove that the solution is optimal to the full program, or find a "violated constraint" in the full problem that is then introduced and the (now strengthened) relaxed program resolved. In this case the subproblem is a variation of the winner determination IP from Section 3, and can be concisely formulated and solved via branch-and-cut.[9]

4.3 Establishing Bounds on Efficiency

Consider some round t in ICE. The round starts with the announcement of prices, denote them π^t, and the provisional trade. The round ends with every bidder having met the δ-MRPAR and ϵ-DIAR activity rules. The question to address is: *what can be established about the efficiency of the trade defined on lower-bound valuations at the end of the round?* It is perhaps unsurprising that MRPAR by itself is sufficient to provide efficiency claims when prices are suitably accurate. What is interesting is that the coupling of MRPAR with DIAR ensures that ICE converges to a provably efficient trade in all cases, with efficiency often established independently of prices by reasoning directly about lower and upper valuation bounds. For the theoretical analysis of convergence to efficiency, we assume *straightforward bidders*, by which we mean *a bidder that always retains his true valuation within the valuation bounds.* (All results could equivalently be phrased in terms of efficiency claims with respect to reported valuations.)

At the closing of each round, ICE makes a determination about whether to move to the last-and-final round. Bidders are notified when this occurs. This last-and-final round

9. The pricing step is the most computationally intensive of all steps in ICE and therefore heavily optimized. In practice, we have found it useful to employ heuristics to seed the set of constraints used in CG. We have also developed algorithmic techniques to speed the search for the appropriate set of constraints in the context of lexicographic refinement: *provisional Locking* of multiple lexicographic values for each CG check, and *lazy constraint checks* in which only a subset of the conditions for CG are routinely checked, even though the complete set is eventually enforced. Please see the technical report at `www.eecs.harvard.edu/~blubin/ice` for complete details of the pricing method.

provides a final opportunity for bidders to update their lower valuation bound information (without exceeding their upper bounds). The exchange finally terminates with the efficient trade and payments determined with respect to the lower valuation bounds: it is these lower bounds that can be considered to be the ultimate bid submitted by each bidder when ICE terminates. Let $\underline{\lambda} \in \arg\max_{\lambda \in \mathcal{F}(x^0)} \sum_i \underline{v}_i(\lambda_i)$ denote the trade that is optimal given the lower bound valuations. As explained in Section 4, ICE is parameterized by target approximation error, Δ^*, providing a lower-bound on the relative efficiency of $\underline{\lambda}$ to the efficient trade λ^* for true valuations. The challenge is to obtain useful bounds on the relative efficiency $\text{EFF}(\underline{\lambda})$ of trade $\underline{\lambda}$. We provide two methods, one of which is price-based and uses duality theory and the second of which directly reasons about the bounds on bidder valuations. We now consider each in turn.

A price-based proof of efficiency. We have already seen in Section 2.2 that a bound on the efficiency of provisional trade λ^α can sometimes be established via prices. This provides a simple method to establish a bound on the efficiency of trade $\underline{\lambda}$. Fix some $\delta \geq 0$. For v^α denoting the provisional valuation profile at the start of round t, and λ^α the corresponding provisional trade, we know that if

(a) bidders meet δ-MRPAR while leaving v^α within their bounds,
(b) prices π^t were δ-approximate EQ prices for v^α and λ^α, and
(c) λ^α is equal to $\underline{\lambda}$, i.e., the efficient trade given the refined lower bound valuations,

then trade $\underline{\lambda}$ is a $2\min(M, \frac{n}{2})\delta$-approximation to the efficient trade λ^* by Theorem 2. We have $\sum_i v_i(\underline{\lambda}_i) + 2\min(M, \frac{n}{2})\delta \geq \sum_i v_i(\lambda_i^*)$, and then,

$$\text{EFF}(\underline{\lambda}) = \frac{\sum_i v_i(\underline{\lambda}_i)}{\sum_i v_i(\lambda^*)} \geq 1 - \frac{2\min(M, \frac{n}{2})}{\sum_i v_i(\lambda_i^*)}\delta \geq 1 - \frac{2\min(M, \frac{n}{2})}{\max_{\lambda \in \mathcal{F}(x^0)} \sum_i \overline{v}_i(\lambda)}\delta, \tag{24}$$

which we define as ω^{price}. Conditioned on (a–c) being met, so that the bound is available, it will satisfy $\omega^{\text{price}} \geq \Delta^*$ for a small enough δ parameter. When the bound is not available we set $\omega^{\text{price}} := 0$.

A direct proof of efficiency. We also provide a complementary, direct, method to establish the relative efficiency of $\underline{\lambda}$ by working with the refined valuation bounds at the end of round t. First, given a bid tree T_i, it is useful to define the ***perturbed valuation with respect to a trade*** λ_i, by assigning the following values to each node β:

$$\tilde{v}_i(\beta) = \begin{cases} \underline{v}_i(\beta) & \text{, if } \beta \in sat_i(\lambda_i) \\ \overline{v}_i(\beta) & \text{, otherwise,} \end{cases} \tag{25}$$

where $\beta \in sat_i(\lambda_i)$ if and only if node β is satisfied given tree T_i and lower bound valuations $\underline{v}_i$ on nodes, and given trade λ_i. The valuation function $\tilde{v}_i$ associated with *TBBL* tree T_i is defined to minimize the value on nodes satisfied by trade λ_i and maximize the value on other nodes. With this concept, and given the valuation bounds, we can now establish the following bound,

$$\text{EFF}(\underline{\lambda}) = \frac{v(\underline{\lambda})}{v(\lambda^*)} \geq \min_{v' \in T, \lambda' \in \mathcal{F}(x^0)} \left[\frac{v'(\underline{\lambda})}{v'(\lambda')}\right] = \min_{\lambda' \in \mathcal{F}(x^0)} \left[\frac{\tilde{v}(\underline{\lambda})}{\tilde{v}(\lambda')}\right] = \frac{\underline{v}(\underline{\lambda})}{\tilde{v}(\tilde{\lambda})}, \tag{26}$$

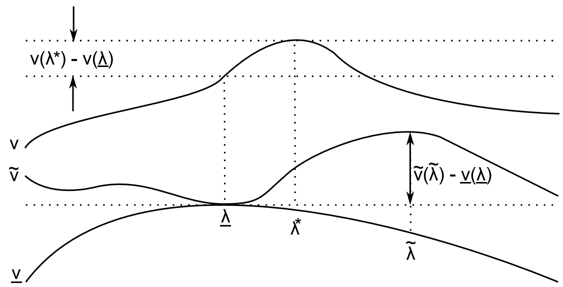


Figure 11: Determining an efficiency bound based on lower and upper valuations.

which we define as ω^{direct}. Notation $\tilde{v} = (\tilde{v}_1, \ldots, \tilde{v}_n)$, and $\tilde{\lambda}$ is the trade that maximizes $\sum_i \tilde{v}_i(\lambda_i)$ across all feasible trades. The first inequality holds because the domain of the minimization includes $v \in T$ and trade $\lambda' = \lambda^*$. The first equality holds because for any $\lambda' \neq \underline{\lambda}$, the worst-case efficiency for $\underline{\lambda}$ occurs when the value $v' \in T$ is selected to minimize the value on nodes $\underline{\lambda} \setminus \lambda'$, maximize the value on nodes $\lambda' \setminus \underline{\lambda}$, and minimize the value on shared nodes, $\lambda' \cap \underline{\lambda}$. Whatever the choice of λ', this valuation is provided through perturbed valuation $\tilde{v}$. For the final equality, $\tilde{v}(\underline{\lambda}) = \underline{v}(\underline{\lambda})$ by definition, and the optimal trade λ' is that which maximizes the value of the denominator, i.e., trade $\tilde{\lambda}$. Figure 11 schematically illustrates the various trades and values used in this bound, and in particular provides some graphical intuition for why $\tilde{v}(\tilde{\lambda}) - \underline{v}(\underline{\lambda}) \geq \tilde{v}(\lambda^*) - \underline{v}(\underline{\lambda}) = \max_{v' \in T}[v'(\lambda^*) - v'(\underline{\lambda})] \geq v(\lambda^*) - v(\underline{\lambda})$.

Combining together. Given the above methods we can establish lower-bound $\omega^{\text{eff}} = \max(\omega^{\text{price}}, \omega^{\text{direct}})$ on the relative efficiency of trade $\underline{\lambda}$. ICE is defined to move to the last-and-final round when either of the following hold:
(a) the error bound $\omega^{\text{eff}} \geq \Delta^*$
(b) there is no trade even at optimistic (i.e., upper-bound) valuations.

Combining this with Theorem 5, we immediately get our main result.

Theorem 6 *When ICE incorporates δ-MRPAR and ϵ-DIAR and when all bidders are straightforward, then the exchange terminates with a trade that is within target approximation error Δ^*, for any $\Delta^* \geq 0$ as $\epsilon \to 0$.*

The use of ϵ-DIAR by itself is sufficient to establish this result. However, it is the use of prices and MRPAR that drives most elicitation in practice, particularly as we fix δ in δ-MRPAR to a tiny constant in actual use. Empirical support for this, along with the quality of the price-based bound and the direct efficiency bounds, is provided in Section 6. For the ϵ parameter in ϵ-DIAR, we find that a simple rule:

$$\epsilon := \frac{1}{2n} \sum_i \sum_{\beta \in T_i} \frac{\overline{v}_{i \in N}(\beta) - \underline{v}_i(\beta)}{|T_i|}, \tag{27}$$

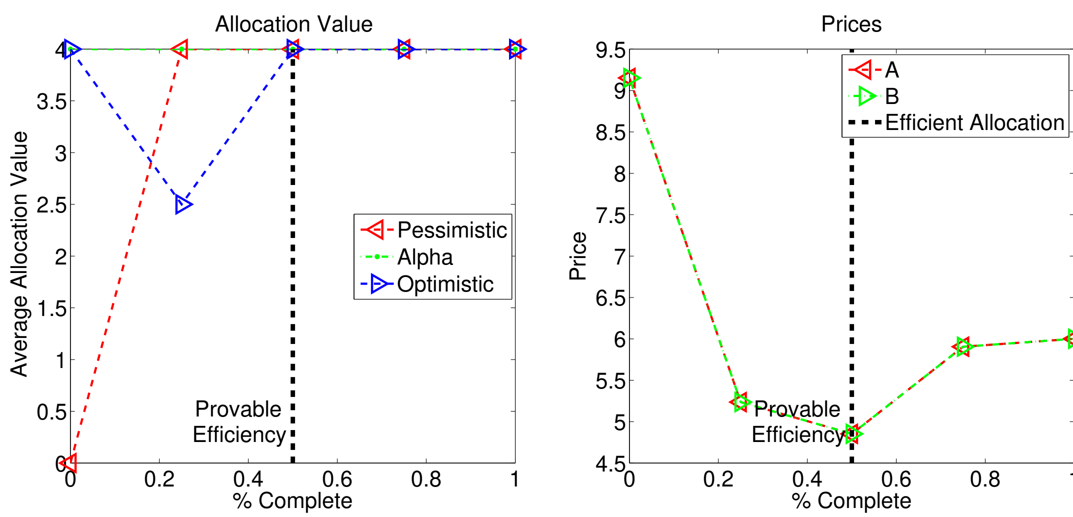
works well. This tends towards zero as more value information is revealed by participants.

One last element of the design of ICE is the precise method by which the provisional valuation profile $v^\alpha = \alpha \underline{v} + (1-\alpha)\overline{v}$ is constructed. This is important because it is then used to determine the provisional trade and price feedback. A simple approach that works well is to define $\alpha := \max(0.5, \omega^{\text{eff}})$. We find that the lower bound of 0.5 is a useful heuristic for early rounds when ω^{eff} is likely to be small, making ICE adopt a provisional valuation in the middle of the valuation bounds when not much is known. The effect is then to push α towards 1 and thus v^{α} towards $\underline{v}$ as the efficiency bound ω^{eff} improves.[10]

5. Illustrative Examples

In this section we illustrate the behavior of the exchange on two simple examples. These examples are provided to give a qualitative feel for its behavior. To construct the examples we populate ICE with very simple, automated bidding agents. These agents use MIP-guided heuristics to minimize the amount of information revealed in the course of passing the activity rules, while maintaining their true value within their lower- and upper-bounds (i.e., they act in a 'straightforward' way). Their reluctance to reveal information models a basic tenet of our design, that it is costly for participants to refine and then reveal information about their values for different trades. A detailed explanation of the operation of these bidding agents is provided in the Appendix.

In this section, and also in presenting our main experimental results, we do *not* move to a last-and-final round. Rather, the bidding agents are programmed to continue to improve their bids past the round at which efficiency is already proved (and when a last-and-final round would ordinarily be declared), and until payments are within some desired accuracy tolerance. We do this to avoid the need to program agents with a strategy for how to bid in the last-and-final round.



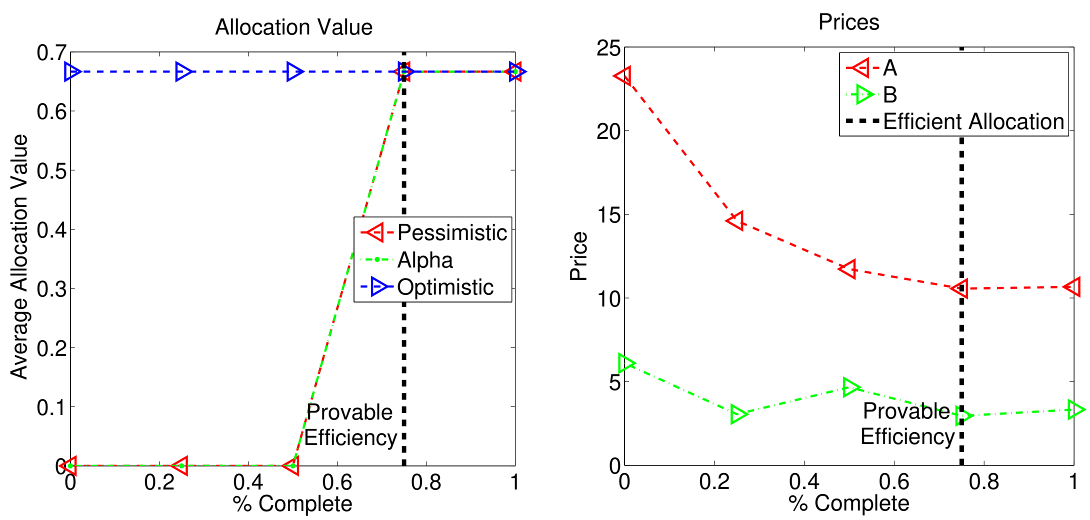
(a) Allocative value

(b) Prices

Figure 12: AgentA: **A** \$8, AgentB: **B** \$8, AgentAB: **A** AND **B** \$10.

10. In some domains, it may also be important to require that *payments* (rather than just the efficiency of trade $\underline{\lambda}$) be accurate enough before moving to the last-and-final round. A bound on payments can be computed in an analogous way to that on efficiency. Whether this is required in practice is likely domain-specific and to depend, for instance, on whether the payments tend to be accurate anyway by the time the trade is approximately accurate, and also on the impact on strategic behavior.

Example 8 *Consider a market with a no-reserve seller of two items **A** and **B**, and three buyers. AgentA demands **A** with a value of \$8, AgentB demands **B** with a value of \$8, and AgentAB demands **A** AND **B** with a value of \$10. Figure 12(a) shows that very quickly the exchange discovers the correct trade. A price between \$5 and \$8 will be accurate in this situation, and we can see that the prices in Figure 12(b) quickly meet this condition. Fairness drives the prices towards \$6, which will be the eventual Threshold payments to AgentA and AgentB. Balance ensures that the prices remain the same for the two items.*



(a) Allocative value (b) Prices

Figure 13: Seller **A** -\$10, Swapper: swap **B** for **A** \$8, Buyer **B** \$4

Example 9 *Consider an example with a Seller offering **A** for a reserve of \$10, a "Swapper" who is willing to pay \$8 if he can swap his **B** for **A**, and a Buyer willing to pay \$4 for **B**. In this more complex example, it takes 4 rounds, as illustrated in Figure 13(a), for a trade to be found in the pessimistic trade. Revelation drives progress towards a completed trade, and as we can see in Figure 13(b), this is reflected in falling prices on the goods. Thus we can see that the price feedback is providing accurate information to the participants: only when the price eventually becomes low enough do the buying bidders actually want a trade to occur– and that is also when the exchange's provisional trade switches. It is also worth noting that the greater valuations the Seller and Swapper place on good **A** result in a net higher price than that for good **B**.*

6. Experimental Analysis

In this section we report the results of a set of experiments that are designed to provide a proof-of-concept for ICE. The results illustrate the scalability of ICE to realistic problem sizes and provide evidence of the effectiveness of the elicitation process and the techniques to bound the efficiency of the provisional trade.

Implementation. First, a brief aside on our experimental implementation. ICE is approximately 20,000 lines of extremely tight Java code, broken up into the functional packages described in Table 1.[11] The prototype is modular so that researchers may easily replace components for experimentation.[12] Because of ICE's complexity, it is essential that the code be constructed in a rigid hierarchy that avoids obscuring the high level logic behind the details of generating, running and integrating the results of MIPs. To this end, the system is written in a series of progressively more abstract "mini-languages" each of which defines a clean, understandable API to the next higher level of logic. Our hierarchy provides a way to hide the extremely delicate steps needed to handle the numerical issues that come out of trying to repeatedly solve coupled optimization problems, where the constraints in one problem may be defined in terms of slightly inaccurate results from an earlier problem. Most of the constraints presented in this paper must be carefully relaxed and monitored in order to handle these numerical precision issues. At the bottom of this hierarchy the MIP specification is fed into our generalized back-end optimization solver interface[13] (we currently support CPLEX and the LGPL-licensed LPSolve), that handles machine load-balancing and parallel MIP/LP solving. This concurrent solving capability is essential, as we need to handle tens of thousands of comparatively simple MIPs/LPs.

Component	**Purpose**	**Lines**
Agent	Strategic behavior and information revelation decisions	2001
Model	XML support to load goods and true valuations	1353
Bidding Language	Implements *TBBL*	2497
Exchange Driver & Communication	Controls exchange, and coordinates agent behavior	1322
Activity/Closing Rule Engines	MRPAR, DIAR and Closing Rules	1830
WD Engine	Logic for WD	685
Pricing Engine	Logic for three pricing stages	1317
MIP Builders	Translates from engines into our optimization APIs	2206
Framework & Instrumentation	Wire components together & Gather data	2642
JOpt	Our Optimization API wrapping CPLEX	2178
Instance Generator	Random Problem Generator	497

Table 1: Exchange components and code breakdown

Experimental set-up. In the experiments, the δ-parameter in MRPAR is set to near zero and both the MRPAR and DIAR activity rule fire in every round. The rule used to define the ϵ-parameter in DIAR is exactly as described in Section 4.1. We adopt the same straightforward bidding agents that were employed in Section 5 (see the Appendix for details). In simulation, we adopt the Threshold payment rule and terminate ICE when the per-agent error in payment relative to the correct payment is within 5% of the average per-agent value for the efficient trade. On typical instances, this incurs an additional 4 rounds beyond those that would be required if we had a last-and-final round. All timing is wall clock time, and does not separately count the large number of parallel threads of execution in the system. The experiments were run on a dual-processor dual-core Pentium

11. Code size is measured in physical source line of code (SLOC).
12. Please contact the authors for access to the source code.
13. `http://www.eecs.harvard.edu/econcs/jopt`

IV 3.2GHz with 8GB of memory and CPLEX 10.1. All results are averaged over 10 trials. The problem instances are available at `http://www.eecs.harvard.edu/~blubin/ice`.

Our instance generator begins by generating a set G of good *types*. Next, for each $j \in G$ it creates $s \geq 1$ copies of each good type, forming a total potential supply in the market of $s|G|$ goods (exactly how many units are in supply depends on the precise structure of bid trees). Each unit is assigned to one of the bidders uniformly at random. The generator creates a bid tree T_i for each bidder by recursively growing it, starting from the root and adopting two phases. For the tree above *depthLow*, each node receives a number of children drawn uniform between *outDegreeLow* and *outDegreeHigh* (a percentage of which are designated as leaves), resulting in an exponential growth in the number of nodes during this phase. By the *width* at some depth we refer to the number of nodes at that depth. Below this point, we carefully control the expected number of children at each node in order to make the expected width conform to a triangle distribution over depth from *depthLow* to *depthMid* to *depthHigh*: we linearly increase the expected width at each depth between *depthLow* and *depthMid* to a fixed multiple (ξ) of the width at *depthLow*, and then linearly decrease the expected width back to zero by *depthHigh*.[14] This provides complex and deep trees without inherently introducing an exponential number of nodes.

Each internal node must be assigned the parameters for its interval choose operator. We typically choose y with a high-triangle distribution between 1 and the number of children and x with a low-triangle distribution between 1 and y. This will bias towards the introduction of IC operators that permit a wide choice in the number of children. Each internal node is also assigned a bonus drawn according to a uniform distribution. Each leaf node is assigned as a "buy" node with a probability $\psi \in [0, 1]$, and then a specific good type for that node is chosen from among those good types for sale in the market. The node is assigned a quantity by drawing from a low-triangle distribution between 1 and the total number in existence.[15] A unit value for the node is then drawn from a specific "buy" distribution, typically uniform, which is multiplied by the quantity and assigned as the node's bonus. The leaf nodes assigned as "sell" nodes have their goods and bonuses determined similarly, this time with goods selected from among those previously assigned to the bidder.[16]

6.1 Experimental Results: Scalability

The first set of results that we present focuses on the computational properties of ICE. Figure 14 shows the runtime performance of the system as we increase the number of bidders while holding all other parameters constant. In this example, 100 goods in 20 types are being traded by bidders with an average of 104 node trees. The graph shows the total wall clock time for all parts of the system. While we see super-linear growth in solve time

14. Note that by setting *depthLow*=*depthMid*=*depthHigh* one can still grow a full tree of a given depth by eliminating phase 2.

15. The total number of goods of a given type in existence may not actually be available for purchase at any price given the structure of seller trees. Thus a bias towards small quantities in "buy" nodes and large quantities in "sell" nodes produces more interesting problem instances.

16. In our experiments, we vary $2 \leq |G| \leq 128$, $1 \leq d \leq 128$, $2 \leq |N| \leq 20$, $2 \leq$ *outDegreeLow* ≤ 8, $2 \leq$ *outDegreeHigh* ≤ 8, $2 \leq$ *depthLow* ≤ 6, $2 \leq$ *depthMid* ≤ 6, $2 \leq$ *depthHigh* ≤ 8, set a balanced buy probability $\psi = 0.5$, and set width multiplier during the second phase to $\xi = 2$. In these examples, buy node bonuses were drawn uniformly from $[10, 100]$, sell nodes bonuses were drawn uniformly from $[-100, -10]$ and internal nodes bonuses uniformly from $[-25, 25]$.

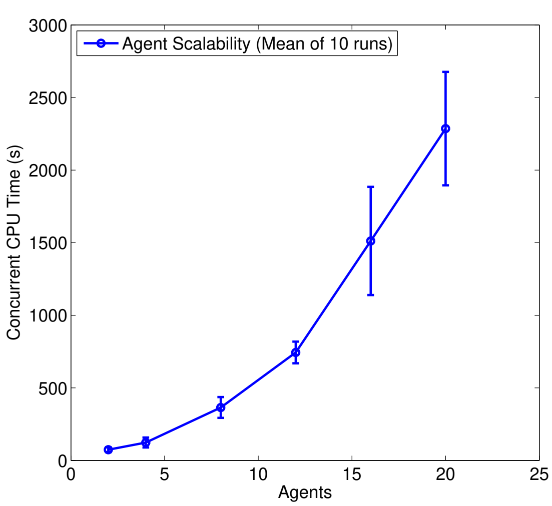


Figure 14: Effect of the number of bidders on the run-time of ICE

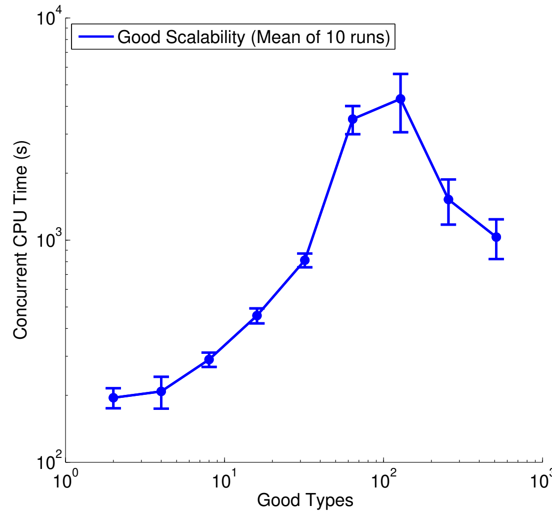


Figure 15: Effect of the number of good types on the run-time of ICE

with the number of bidders, the constants of this growth are such that markets with large numbers of bidders can be efficiently solved (solving for 20 bidders in around 40 minutes). The error bars in all plots are for the standard error of the statistic.

In Figure 15 we can see the effect of varying the number of types of goods (retaining 5 units of each good in the supply) on computation time. For this example we adopt 10 bidders, and the same tree generation parameters. A likely explanation for eventual concavity of the run-time performance is suggested by the decrease in the average (item) price upon termination of ICE as the number of types of goods are increased (see Figure 16). The average price provides a good proxy for the competitiveness of the market. Adding

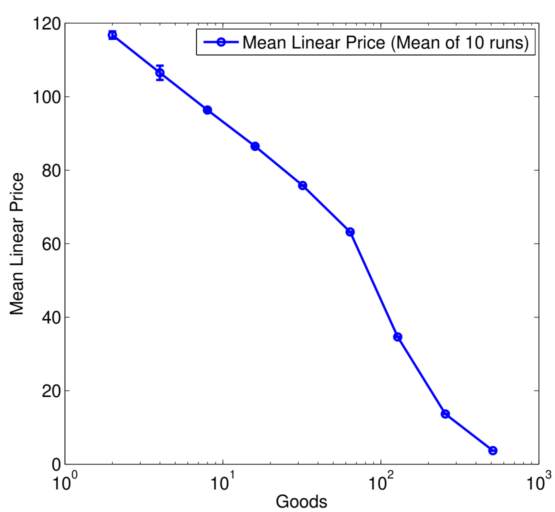


Figure 16: Effect of the number of goods on the average item price upon termination of ICE.

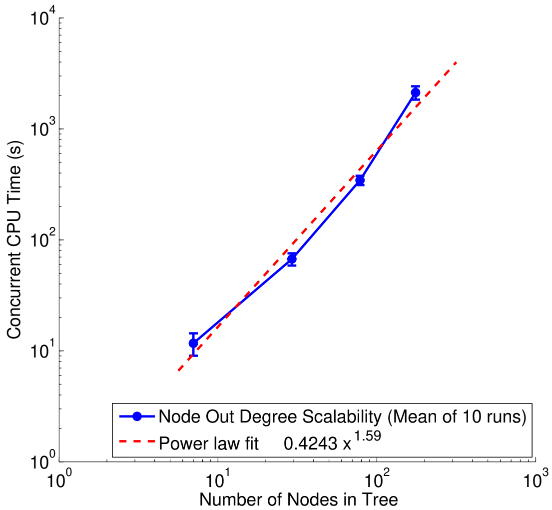


Figure 17: Effect of bid-tree size on run-time of ICE: Varying the node-out degree.

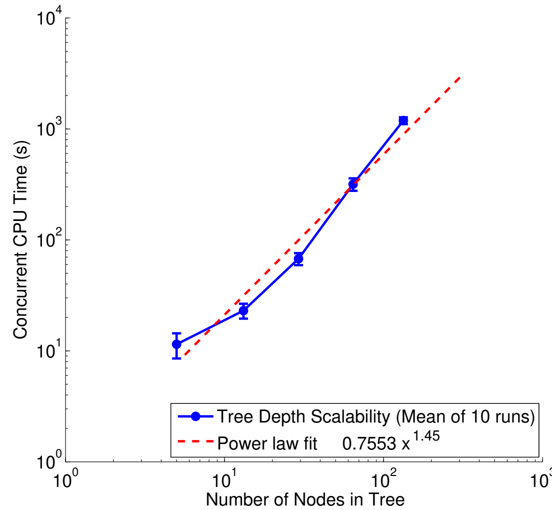


Figure 18: Effect of bid-tree size on run-time of ICE: Varying the tree depth.

goods to the problem will initially make the winner determination problem more difficult, but only until there is a large over-supply, at which point the outcome is easier to determine.

Figures 17 and 18 illustrate the change in run time with the size of bid trees. Here we use only the first phase of our tree-generator to avoid confounding the effects of size with structural complexity. In both experiments, 100 goods in 20 types were being traded by 10 bidders. In Figure 17 we vary the number of children of any given node while in Figure 18 we vary the depth of the tree. Increasing the branching factor and/or tree depth results in an exponential growth in tree size, which necessarily corresponds to an exponential growth in runtime. However, if we account for this by instead plotting against the number of nodes in the trees, we see that both graphs indicate a *near-polynomial increase in runtime with tree size*. We fit a polynomial function to this data of the form $y = Ax^b$, indicating that this growth is approximately of degree 1.5 in the range of tree sizes considered in these experiments.

6.2 Empirical Results: Economic Properties

The second set of results that we present focus on the economic properties of ICE: the efficiency of trade across rounds, the effectiveness of preference elicitation, and the accuracy and stability of prices. For this set of experiments we average over 10 problem instances, each with 8 bidders, a potential supply of 100 goods in 20 types, and bid trees with an average of 104 nodes.

Figure 19 plots the *true* efficiency of the trades computed at pessimistic (lower bounds $\underline{v}$), provisional (α-valuation v^{α}) and optimistic (upper bounds $\overline{v}$) valuations across rounds. In this graph and those that follow, the x-axis indicates the number of rounds completed as a percentage of the total number of rounds until termination which enables results to be aggregated across multiple instances, each of which can have a different number of total

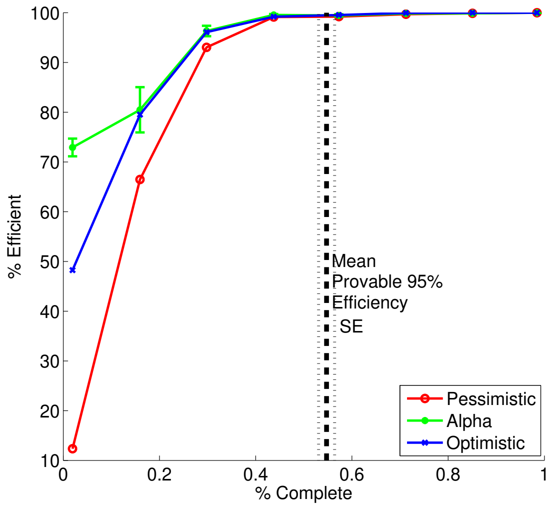


Figure 19: Efficiency of the optimistic, provisional, and pessimistic trades across rounds.

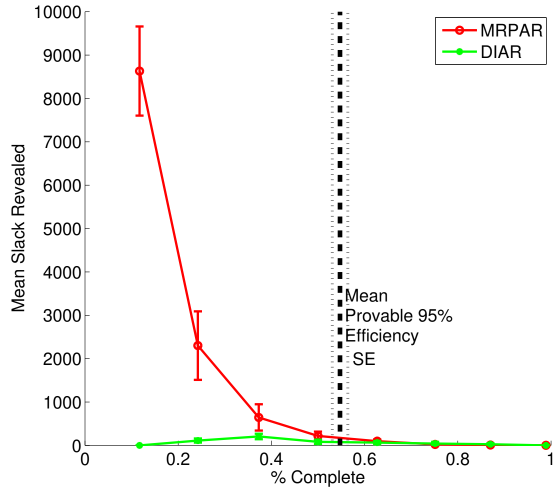


Figure 20: Average reduction in value uncertainty due to each rule.

rounds.[17] The vertical (dashed) line indicates the average percentage complete when the trade is provably 95% efficient. The exchange remains open past this point while payments converge (and because we simulate the outcome of the last-and-final round by continuing progress with our straightforward bidding agents). The two lines on either side represent one standard error of this statistic.

In Figure 19, we see that the exchange quickly converges to highly efficient trades, taking an average of 6.8 rounds to achieve efficiency. In general, the optimistic trade (i.e., computed from upper bounds $\overline{v}$) has higher (true) efficiency than the pessimistic one (i.e., computed from lower bounds $\underline{v}$), while the efficiency of the provisional trade λ^α is typically better than both. This justifies the design decision to adopt the provisional valuations and provisional trade in driving the exchange dynamics. It also suggests that exchanges with the traditional paradigm of improving bids (i.e., increasing lower bound claims on valuations) would allow little useful feedback in early rounds: the efficiency of the pessimistic trade—all that would be available without information about the upper-bounds of bidder valuations—is initially very poor.

Figure 20 shows the average amount of revelation caused by MRPAR and DIAR in each round of ICE. Revelation is measured here in terms of the *absolute tightening of upper and lower bounds, summed across the bid trees*. The MRPAR activity rule is the main driving force behind the revelation of information and the vast majority of revelation (in absolute terms) occurs within the first 25% of rounds. DIAR plays a role in making progress towards identifying the efficient trade but only once MRPAR has substantially reduced the value uncertainty and despite firing in every round. One can think of MRPAR as our rocket's main engine, and DIAR as a thruster for mid-course correction. ICE determines the efficient

17. Each data point represents the average across the 10 instances, and is determined by averaging the underlying points in its neighborhood. Error-bars indicate the standard error (SE) of this mean. Thus, the figures are essentially a histogram rendered as a line graph.

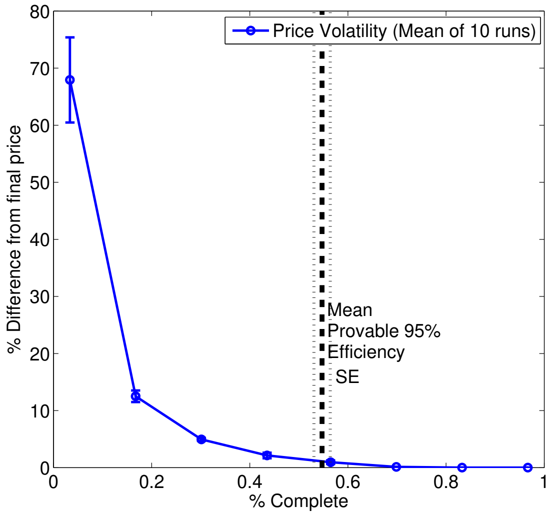


Figure 21: Price trajectory: Closeness of prices in each round to the final prices

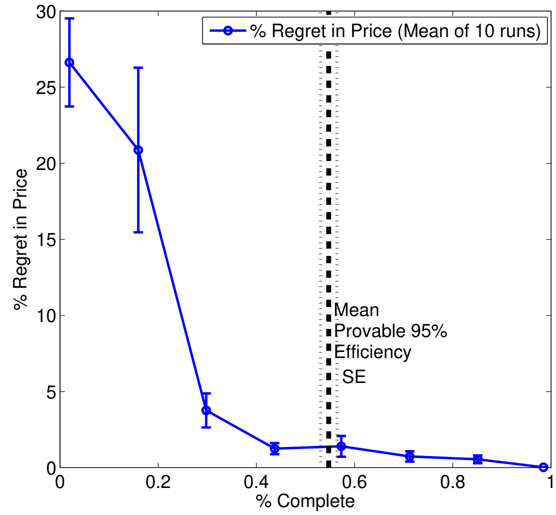


Figure 22: Regret in best-response by bidders due to price inaccuracy relative to final prices.

trade when the average node in a *TBBL* tree still retains a gap between the upper and lower bounds on value at the node equal to around 62% of the maximum (true) value that a node could contribute to a bidder's value, roughly the maximum marginal value contributed by a node over all feasible trades. We see that ICE is successful in directing preference elicitation to information that is relevant to determining the efficient trade.

We now provide two different views on the effectiveness of prices. Figure 21 shows the mean percentage absolute difference between the prices computed in some round and the prices computed in the final round. Prices quickly converge. In our experiments we have driven the exchange beyond the efficient solution in order to converge to the Threshold payments, but we see that most of the price information is already available at the point of efficiency. Figure 22 provides information about the quality of the price feedback. We plot the 'regret', averaged across bidders and runs, from the best-response trade as determined from intermediate prices in comparison to the best-response to final prices, where the regret is defined in terms of lost payoff at those final prices. Define the regret to bidder i for his best response $\lambda'_i = \arg\max_{\lambda_i \in \mathcal{F}_i(x^0)}[v_i(\lambda_i) - p^{\hat{\pi}}(\lambda_i)]$, to prices $\hat{\pi}$, given that the final prices are π, as:

$$\text{Regret}_i(\lambda'_i, \pi) = \left(1 - \frac{v_i(\lambda'_i) - p^{\pi}(\lambda'_i)}{\max\limits_{\lambda_i \in \mathcal{F}_i(x^0)} v_i(\lambda_i) - p^{\pi}(\lambda_i)}\right) \times 100\%. \tag{28}$$

As the payoff from trade λ'_i, when evaluated at prices π, approaches that from the best-response trade at prices π, then $\text{Regret}_i(\lambda'_i, \pi)$ approaches 0%. Figure 22 plots the *average regret across all bidders* as a function of the number of rounds completed in ICE. The regret is low: 11.2% when averaged across all rounds before the efficient trade is determined and 7.0% when averaged across all rounds. That regret falls across rounds also shows that prices become more and more informative as the rounds proceed.

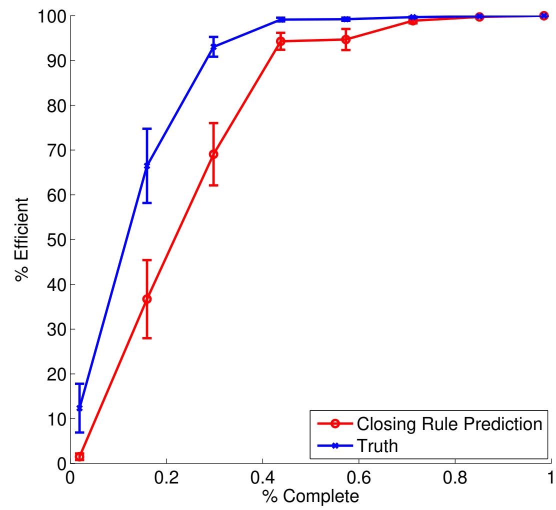


Figure 23: Comparison between the actual efficiency of the pessimistic trade and the ω^{direct} bound.

Finally, we present experimental results that relate to the two methods that ICE employs to bound the final efficiency of the pessimistic trade. The total pricing error across all bidders in each round as determined within pricing in terms of $(\lambda^\alpha, v^\alpha)$, and normalized here by the total *true* value of the efficient trade, is already small (at 8.5%) in initial rounds and falls to around 3% by final rounds of ICE. This suggests that a price-based bound is quite informative, although note that this is defined in terms of the error given $(\lambda^\alpha, v^\alpha)$ and does not immediately map to a price-based accuracy claim for true valuations and for the current trade $\underline{\lambda}$ defined on lower bound valuations. Figure 23 compares the actual efficiency of the pessimistic trade $\underline{\lambda}$ in each round with that estimated by the ω^{direct} bound on efficiency that is available to the exchange. This confirms that the direct bound is reasonably tight, and very effective in bounding the true efficiency regardless of the accuracy of the prices.

7. Conclusions

In this work we designed and implemented a scalable and highly expressive iterative combinatorial exchange. The design includes many interesting features, including: a new tree-based language for combinatorial exchange environments, a new method to construct approximate linear prices from expressive languages, a proxied architecture with optimistic and pessimistic valuations coupled with price-based activity rules to drive preference elicitation, and a direct method to estimate the final efficiency of the trade in terms of valuation bounds. By adopting proxy agents that receive direct, expressive claims on upper and lower valuations bounds we are able to form claims about efficiency despite using only linear prices. These bounds also allow for good progress in early rounds, and even when there is no efficient trade at lower bound (pessimistic) values. Experimental results with automated, simple bidding agents indicate good behavior in terms of both scalability and economic properties.

There are many intriguing opportunities for future work. It will be especially interesting to instantiate special-cases of the ICE design to domains for which there exist strategyproof, static (two-sided) combinatorial market designs. This would bring straightforward bidding strategies into an *ex post* Nash equilibrium. For example, it should be possible to integrate methods such as trade-reduction (McAfee, 1992) and its generalizations (Babaioff & Walsh, 2005; Chu & Shen, 2007) in domains with restricted expressiveness. We can also consider ICE as a combinatorial *auction* rather than exchange, where a direct appeal to VCG payments would provide incentive compatibility. The other two major directions for future work are to: (a) modify the design to allow bidders to refine the *structure*, not just the valuation bounds on their *TBBL* tree, across rounds; (b) extend ICE to work in a dynamic environment with a changing bidder population, for instance maintaining linear price feedback and periodically clearing. Recent progress in on-line mechanism design includes truthful, dynamic double auctions for very simple expressiveness (Blum, Sandholm, & Zinkevich, 2006; Bredin, Parkes, & Duong, 2007), but does not extend to the kind of expressiveness and price sophistication present in ICE; see the work of Parkes (2007) for a recent survey. Lastly, the incentive properties of ICE are very much dependent on the payment rule used which argues for further analysis of the Threshold rule and its alternatives.

Acknowledgments

This work is supported in part by NSF grant IIS-0238147. An earlier version of this paper appeared in the Proc. 6th ACM Conference on Electronic Commerce, 2005. The *TBBL* language is also described in a workshop paper (Cavallo et al., 2005). The primary authors of this paper are Benjamin Lubin, Adam Juda and David Parkes. Thanks to the anonymous reviewers and associate editor of JAIR for extremely helpful comments. Nick Elprin, Loizos Michael and Hassan Sultan contributed to earlier versions of this work. Our thanks to the students in Al Roth's class (Econ 2056) who participated in a trial of this system and to Cynthia Barnhart for airline domain expertise. We would also like to thank Evan Kwerel and George Donohue for early motivation and encouragement. Some of the computation used in the preparation of this manuscript was performed on the Crimson Grid in the Harvard School of Engineering and Applied Sciences. Finally, this paper's genesis is from CS 286r "Topics at the Interface of Computer Science and Economics" taught at Harvard in Spring 2004. Thanks to all the students for their many early, innovative ideas.

Appendix A. Computation for MRPAR

In this section we show that MRPAR can be computed by solving a sequence of 3 MIPs. We begin by considering the special case of $\delta = 0$. The general case follows almost immediately. Define a *candidate passing trade*, λ_i^L, as:

$$\lambda_i^L \in \arg \max_{\lambda_i \in \mathcal{F}_i(x^0)} \underline{v}_i(\lambda_i) - p^\pi(\lambda_i) \tag{29}$$

breaking ties

(i) to maximize $\overline{v}_i(\lambda_i) - \underline{v}_i(\lambda_i)$

(ii) in favor of λ_i^α

This can be computed by solving one MIP to maximize $\underline{v}_i(\lambda_i) - p^\pi(\lambda_i)$, followed by a second MIP in which this objective is incorporated as a constraint and $\overline{v}_i(\lambda_i) - \underline{v}_i(\lambda_i)$ becomes the objective. Given perturbed valuation $\tilde{v}_i$, defined with respect to trade λ_i^L (as in Section 4), we can define a *witness trade*, λ_i^U, as:

$$\lambda_i^U \in \arg \max_{\lambda_i \in \mathcal{F}_i(x^0)} \tilde{v}_i(\lambda_i) - p^\pi(\lambda_i). \tag{30}$$

This can be found by solving a third MIP. Given prices π, provisional trade λ_i^α and bid tree T_i, the *computational* MRPAR rule (C-MRPAR) for the case of $\delta = 0$ can now be defined as:

(1) $\underline{v}_i(\lambda_i^L) - p^\pi(\lambda_i^L) \geq \tilde{v}_i(\lambda_i^U) - p^\pi(\lambda_i^U)$,**and**

(2) $\lambda_i^L = \lambda_i^\alpha$, **or** $\underline{v}_i(\lambda_i^L) - p^\pi(\lambda_i^L) > \tilde{v}_i(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha)$

We now establish that C-MRPAR is equivalent to MRPAR, as defined by (19)–(21).

Lemma 4 *Given trades λ_i and λ_i', prices π, and tree T_i, we have $\theta_i^\pi(\lambda_i, \lambda_i', v_i') \geq 0$, $\forall v_i' \in T_i$ if and only if $\underline{v}_i(\lambda_i) - p^\pi(\lambda_i) \geq \tilde{v}_i(\lambda_i') - p^\pi(\lambda_i')$, where $\tilde{v}_i$ is defined with respect to trade λ_i.*

Proof: Direction ($\Rightarrow$) is immediate since $\tilde{v}_i \in T_i$. Consider direction ($\Leftarrow$) and suppose, for contradiction, that $\underline{v}_i(\lambda_i) - p^\pi(\lambda_i) \geq \tilde{v}_i(\lambda_i') - p^\pi(\lambda_i')$ but there exists some $v_i' \in T_i$ such that $v_i'(\lambda_i) - p^\pi(\lambda_i) < v_i'(\lambda_i') - p^\pi(\lambda_i')$. Subtract $\sum_{\beta \in \lambda_i \cap \lambda_i'} [v_i'(\beta) - \underline{v}_i(\beta)]$ from both sides, where $\beta \in \lambda_i$ indicates that node β is satisfied by trade λ_i, to get

$$\sum_{\beta \in \lambda_i \setminus \lambda_i'} v_i'(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} v_i'(\beta) - \sum_{\beta \in \lambda_i \cap \lambda_i'} v_i'(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} \underline{v}_i(\beta) - p^\pi(\lambda_i) <$$

$$\sum_{\beta \in \lambda_i' \setminus \lambda_i} v_i'(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} v_i'(\beta) - \sum_{\beta \in \lambda_i \cap \lambda_i'} v_i'(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} \underline{v}_i(\beta) - p^\pi(\lambda_i') \tag{31}$$

$$\Rightarrow \sum_{\beta \in \lambda_i \setminus \lambda_i'} \underline{v}_i(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} \underline{v}_i(\beta) - p^\pi(\lambda_i) < \sum_{\beta \in \lambda_i' \setminus \lambda_i} \overline{v}_i(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} \underline{v}_i(\beta) - p^\pi(\lambda_i') \tag{32}$$

$$\Rightarrow \quad \underline{v}_i(\lambda_i) - p^\pi(\lambda_i) < \tilde{v}_i(\lambda_i') - p^\pi(\lambda_i'), \tag{33}$$

which is a contradiction. □

Lemma 5 *Given trade λ_i, prices π, and tree T_i then $\theta_i^\pi(\lambda_i, \lambda_i', v_i') \geq 0$, $\forall v_i' \in T_i$, $\forall \lambda_i' \in \mathcal{F}_i(x^0)$, if and only if $\underline{v}_i(\lambda_i) - p^\pi(\lambda_i) \geq \tilde{v}_i(\lambda_i^U) - p^\pi(\lambda_i^U)$, where $\tilde{v}_i$ is defined with respect to trade λ_i and λ_i^U is the witness trade.*

Proof: Direction ($\Rightarrow$) is immediate since $\tilde{v}_i \in T_i$ and $\lambda_i^U \in \mathcal{F}_i(x^0)$. Consider direction ($\Leftarrow$) and suppose, for contradiction, that $\underline{v}_i(\lambda_i) - p^\pi(\lambda_i) \geq \tilde{v}_i(\lambda_i^U) - p^\pi(\lambda_i^U)$ but there exists some $\lambda_i' \in \mathcal{F}_i(x^0)$ and $v_i' \in T_i$ such that $\theta_i^\pi(\lambda_i, \lambda_i', v_i') < 0$. By Lemma 4, this means $\underline{v}_i(\lambda_i) - p^\pi(\lambda_i) < \tilde{v}_i(\lambda_i') - p^\pi(\lambda_i')$. But, we have a contradiction because

$$\underline{v}_i(\lambda_i) - p^\pi(\lambda_i) \geq \tilde{v}_i(\lambda_i^U) - p^\pi(\lambda_i^U) \tag{34}$$

$$= \max_{\lambda_i'' \in \mathcal{F}_i(x^0)} \tilde{v}_i(\lambda_i'') - p^\pi(\lambda_i'') \geq \tilde{v}_i(\lambda_i') - p^\pi(\lambda_i') \tag{35}$$

□

Theorem 7 *C-MRPAR is equivalent to δ-MRPAR for $\delta = 0$.*

Proof: Comparing (17) and (18) with C-MRPAR, and given Lemmas 4 and 5, all that is left to show is that it is sufficient to check λ_i^L, as the only candidate to pass MRPAR. That is, we need to show that if there is some $\breve{\lambda}_i \in \mathcal{F}_i(x^0)$ that satisfies MRPAR then λ_i^L satisfies MRPAR. We argue as follows:

1. Trade $\breve{\lambda}_i$ must solve $\max_{\lambda_i \in \mathcal{F}_i(x^0)}[\underline{v}_i(\lambda_i) - p^\pi(\lambda_i)]$. Otherwise, there is some λ_i' with $\underline{v}_i(\lambda_i') - p^\pi(\lambda_i') > \underline{v}_i(\breve{\lambda}_i) - p^\pi(\breve{\lambda}_i)$. A contradiction with (17).

2. Trade $\breve{\lambda}_i$ must also break ties in favor of maximizing $\overline{v}_i(\lambda_i) - \underline{v}_i(\lambda_i)$. Otherwise, there is some λ_i' with the same profit as $\breve{\lambda}_i$ at $\underline{v}_i$, with $\overline{v}_i(\lambda_i') - \underline{v}_i(\lambda_i') > \overline{v}_i(\breve{\lambda}_i) - \underline{v}_i(\breve{\lambda}_i)$. This implies $\overline{v}_i(\lambda_i') - \overline{v}_i(\breve{\lambda}_i) > \underline{v}_i(\lambda_i') - \underline{v}_i(\breve{\lambda}_i)$, and $\theta_i^\pi(\lambda_i', \breve{\lambda}_i, \overline{v}_i) > \theta_i^\pi(\lambda_i', \breve{\lambda}_i, \underline{v}_i)$. But, since λ_i' has the same profit as $\breve{\lambda}_i$ at $\underline{v}_i$ we have $\theta_i^\pi(\lambda_i', \breve{\lambda}_i, \underline{v}_i) = 0$ and so $\theta_i^\pi(\lambda_i', \breve{\lambda}_i, \overline{v}_i) > 0$. This is a contradiction with (17).

3. Proceed now by case analysis. Either $\breve{\lambda}_i = \lambda_i^\alpha$, in which case we are done because this will be explicitly selected as candidate passing trade λ_i^L. For the other case, let Λ_i^L denote all feasible solutions to (29) and consider the difficult case when $|\Lambda_i^L| > 1$. We argue that if $\breve{\lambda}_i \in \Lambda_i^L$ satisfies MRPAR, then so does any other trade $\lambda_i' \in \Lambda_i^L$, with $\lambda_i' \neq \breve{\lambda}_i$. By MRPAR, we have $\theta_i^\pi(\breve{\lambda}_i, \lambda_i', v_i') \geq 0$, $\forall v_i' \in T_i$. In particular, $\tilde{v}_i(\breve{\lambda}_i) - p^\pi(\breve{\lambda}_i) \geq \tilde{v}_i(\lambda_i') - p^\pi(\lambda_i')$, where $\tilde{v}_i$ is defined with respect to $\breve{\lambda}_i$, and equivalently,

$$\underline{v}_i(\breve{\lambda}_i) - p^\pi(\breve{\lambda}_i) \geq \tilde{v}_i(\lambda_i') - p^\pi(\lambda_i'). \tag{36}$$

   On the other hand,

$$\underline{v}_i(\breve{\lambda}_i) - p^\pi(\breve{\lambda}_i) = \underline{v}_i(\lambda_i') - p^\pi(\lambda_i'), \tag{37}$$

   since both are in Λ_i^L. Taking (36) together with (37), we must have that λ_i' satisfies no uncertain value nodes in T_i not also satisfied in $\breve{\lambda}_i$. Moreover, since $\overline{v}_i(\breve{\lambda}_i) - \underline{v}_i(\breve{\lambda}_i) = \overline{v}_i(\lambda_i') - \underline{v}_i(\lambda_i')$, both trades must satisfy *exactly* the same uncertain value nodes. Finally, by (37) the profit from all fixed value nodes in T_i must be the same in both trades. We conclude that the profit is the same for *all* $v_i' \in T_i$ for $\breve{\lambda}_i$ and λ_i' at the current prices and MRPAR is satisfied by either trade.

□

To understand the importance of the tie-breaking rule (i) in selecting the candidate passing trade, λ_i^L, in C-MRPAR, consider the following example for MRPAR with $\delta = 0$:

Example 10 *A bidder has $XOR(+A, +B)$ and a value of 5 on the leaf $+A$ and a value range of [5,10] on leaf $+B$. Suppose prices are currently 3 for each of A and B and $\lambda_i^\alpha = +B$. The MRPAR rule is satisfied because the market knows that however the remaining value uncertainty on $+B$ is resolved the bidder will always (weakly) prefer $+B$ to $+A$ and $+B$ is λ_i^α. Notice that both $+A$ and $+B$ have the same pessimistic utility, but only $+B$ can satisfy MRPAR. But $+B$ has maximal value uncertainty, and therefore this is selected over $+A$ by C-MRPAR.*

To understand the importance of selecting, and evaluating, λ_i^U with respect to $\tilde{v}_i$ rather than $\overline{v}_i$, consider the following example (again for $\delta = 0$). It illustrates the role of "shared uncertainty" in the tree, which occurs when multiple trades share a node with uncertain value and the value, although uncertain, will be resolved in the same way for both trades.

Example 11 *A bidder has XOR(+A, +B) and value bounds* $[5, 10]$ *on the* root *node and a value of 1 on leaf +A. Suppose prices are currently 3 for each of A and B and* $\lambda_i^\alpha = +B$*. The MRPAR rule is satisfied because the bidder strictly prefers +A to +B, whichever way the uncertain value on the root node is ultimately resolved. C-MRPAR selects* λ_i^L *as "buy A", with payoff* $\underline{v}_i(\lambda_i^L) - p^\pi(\lambda_i^L) = 5 + 1 - 3 = 3$*. At valuation* $\overline{v}_i$*, the witness trade "buy B" would be selected and have payoff* $10 - 3 = 7$ *and seem to violate MRPAR. But, whichever way the uncertain value at the root is resolved it will affect +A and +B in the same way. This is addressed by setting* $\tilde{v}_i(\beta) = \underline{v}_i(\beta) = 5$ *on the root node, the same value adopted in determining the payoff from* λ_i^L*. Evaluated at* $\tilde{v}_i$*, the witness is "buy A" and (1) of C-MRPAR is trivially satisfied while (2) is satisfied since* $3 > 5 - 3 = 2$*.*

For δ-MRPAR with $\delta > 0$, we adopt a slight variation, with a δ-C-MRPAR procedure defined as:

(1) Check $\theta_i^\pi(\lambda_i^\alpha, \lambda_i', v_i') \geq -\delta$ for all $v_i' \in T_i$, all $\lambda_i' \in \mathcal{F}_i(x^0)$ directly, by application of Lemma 5 with valuation $\tilde{v}_i$ defined with respect to trade λ_i^α, and test

$$\underline{v}_i(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha) \geq \tilde{v}_i(\lambda_i^U) - p^\pi(\lambda_i^U) - \delta \tag{38}$$

(2) If this is not satisfied then fall back on C-MRPAR to verify (20) and (21), with candidate passing trade λ_i^L modified from (29) to drop tie-breaking in favor of λ_i^α and with the second step of C-MRPAR modified to require $\underline{v}_i(\lambda_i^L) - p^\pi(\lambda_i^L) > \tilde{v}_i(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha) + \delta$, again with $\tilde{v}_i$ defined with respect to λ_i^L.

The argument adopted in the proof of Theorem 7 remains valid in establishing that it is sufficient to consider λ_i^L, as defined in δ-C-MRPAR, in the case that λ_i^α does not pass the activity rule.

Appendix B. Computation for DIAR

The ϵ-DIAR rule can be verified by solving two MIPs. The first optimization problem identifies the trade with maximal DIAR error for which the current bounds refinement has improved this error by at least ϵ:

$$\begin{aligned}
\Delta_i^P = & \max_{\lambda_i \in \mathcal{F}_i(x^0)} [\tilde{v}_i^0(\lambda_i) - p^\pi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))] && (39)\\
& \text{s.t.} \quad (\tilde{v}_i^0(\lambda_i) - p^\pi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))) \\
& \qquad - (\tilde{v}_i^1(\lambda_i) - p^\pi(\lambda_i) - (\underline{v}_i^1(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))) \geq \epsilon && (40)\\
= & -C + \max_{\lambda_i \in \mathcal{F}_i(x^0)} \tilde{v}_i^0(\lambda_i) - p^\pi(\lambda_i) && (41)\\
& \text{s.t.} \quad \tilde{v}_i^0(\lambda_i) - \underline{v}_i^0(\lambda_i^\alpha) - \tilde{v}_i^1(\lambda_i) + \underline{v}_i^1(\lambda_i^\alpha) \geq \epsilon, && (42)
\end{aligned}$$

where $\tilde{v}_i^0$ and $\tilde{v}_i^1$ are defined with respect to λ_i^α, v^0 and v^1 represent valuations defined before and after the bidder's refinement respectively, and $C = \underline{v}_i^0(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha)$. Note that the problem could be infeasible, in which case we define $\Delta_i^P := -\infty$.

The second optimization identifies the trade with maximal DIAR error for which v^1 still allows for the possibility of valuation bounds that provide an ϵ error reduction over v^0:

$$\begin{aligned}
\Delta_i^F = \max_{\lambda_i \in \mathcal{F}_i(x^0)} & [\tilde{v}_i^0(\lambda_i) - p^\pi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))] && (43)\\
\text{s.t.} \quad & (\tilde{v}_i^0(\lambda_i) - p^\pi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))) \\
& - (\underline{v}_i^1(\lambda_i) - p^\pi(\lambda_i) - (\breve{v}_i^1(\lambda_i^\alpha) - p^\pi(\lambda_i^\alpha))) \geq \epsilon && (44)\\
= -C + \max_{\lambda_i \in \mathcal{F}_i(x^0)} & \tilde{v}_i^0(\lambda_i) - p^\pi(\lambda_i) && (45)\\
\text{s.t.} \quad & \tilde{v}_i^0(\lambda_i) - \underline{v}_i^0(\lambda_i^\alpha) - \underline{v}_i^1(\lambda_i) + \breve{v}_i^1(\lambda_i^\alpha) \geq \epsilon, && (46)
\end{aligned}$$

where $\tilde{v}_i$ is defined with respect to λ_i^α, and $\breve{v}_i$ is similarly defined with respect to λ_i. The second term in (44) recognizes that it remains possible to decrease the value on λ_i to the new lower-bound $\underline{v}_i^1(\lambda_i)$, while increasing the value on λ_i^α to the new upper-bound $\overline{v}_i^1(\lambda_i^\alpha)$ except on those nodes that are shared with λ_i, giving $\breve{v}_i^1(\lambda_i^\alpha)$. We see that (46) is equivalent to:

$$\sum_{\beta \in \lambda_i \setminus \lambda_i^\alpha} [\overline{v}_i^0(\beta) - \underline{v}_i^1(\beta)] + \sum_{\beta \in \lambda_i^\alpha \setminus \lambda_i} [\overline{v}_i^1(\beta) - \underline{v}_i^0(\beta)] \geq \epsilon,$$

which calculates the amount of refinement that is still possible in service of reducing the DIAR error. Note the problem could be infeasible, in which case we define $\Delta_i^F := -\infty$. *We ultimately compare the two solutions, and the bidder passes DIAR if and only if* $\Delta_i^P \geq \Delta_i^F$.

Appendix C. The Automated Bidding Agents and Bidder Feedback

The bidding agents that are used for the simulation experiments are designed to minimize the amount of information revealed in order to pass the activity rules and while remaining straightforward so that the true valuation is consistent with lower and upper valuations. In summarizing the behavior of the bidding agents, there are three things to explain: (a) the method that we adopt in place of the last-and-final round; (b) the feedback that is provided by ICE to bidders in meeting MRPAR and DIAR; and (c) the logic that is followed by the bidding agents. Rather than define a method for bidding agents to adjust their bounds in a last-and-final round, we keep ICE open in simulation past the point in which it would ordinarily go to last-and-final. Past this point, the bidding agents continue to refine their bounds and ICE terminates when the payments are within some desired accuracy. Each bidding agent in this phase reduces its uncertainty by some multiplicative factor on all nodes that are active in the current provisional trade or in any of the provisional trades for the economies with bidder i removed. This is adopted for simulation purposes only.

Our bidding agents operate in a loop, heuristically modifying their valuation bounds in trying to meet MRPAR and DIAR and querying the proxy for advice. The proxy provides guidance to help the bidding agent further refine its valuation so it can meet the activity rule. For both MRPAR and DIAR, the optimization problems that are solved in checking whether a bidder has satisfied the activity rule also provide information that can guide the bidder. First consider MRPAR and recall that λ_i^L is the candidate passing trade and λ_i^U is the witness trade. The following lemma is easy, and stated without proof:

Lemma 6 *When MRPAR is not satisfied for the current valuation bounds, a bidder must increase a lower bound on at least one node in $\{\lambda_i^L \setminus \lambda_i^U\}$, or decrease an upper bound on at least one node in $\{\lambda_i^U \setminus \lambda_i^L\}$, in order to meet the activity rule.*

Once a simple bidder makes some changes on some subset of these nodes, the bidder can inquire if he has passed the activity rule. The proxy can then respond "yes" or can revise the set of nodes on which the bidding agent should refine its valuation bounds. A similar functionality is provided for DIAR. This time the trade that solves the second MIP (with DIAR error Δ_i^F) is provided as feedback, together with information about how much the bidder must either further reduce the error, or further constrain the possibilities on this trade, to satisfy DIAR. The bidding agent can determine from this information which nodes it must modify, and by how much in total, and is free to decide how much to modify each node to satisfy the rule. The key to our agent design is the following lemma:

Lemma 7 *The trade with which a straightforward bidder passes MRPAR (for $\delta = 0$) must be a trade that is weakly preferred by the bidder to all other trades for his true valuation.*

Proof: By contradiction. Suppose true valuation $v_i \in T_i$ and trade $\breve{\lambda}_i$ meets MRPAR but is not a weakly preferred trade at the true valuation and prices π. Then, there exists a trade $\lambda_i^* \in \mathcal{F}_i(x^0)$ such that $\theta_i^\pi(\lambda_i^*, \breve{\lambda}_i, v_i) > 0$. But, this is a contradiction with MRPAR since $\theta_i^\pi(\breve{\lambda}_i, \lambda_i', v_i') \geq 0$ for all $v_i' \in T_i$ and all $\lambda_i' \in \mathcal{F}_i(x^0)$, including $v_i' = v_i$ and $\lambda_i' = \lambda_i^*$. □

We use this observation to define a procedure UpdateMRPAR by which a bidder can intelligently refine its valuation bounds to meet MRPAR. Let $\breve{\lambda}_i$ be the trade with which we hope to pass MRPAR, and define $u_i(\lambda_i, \pi) = v_i(\lambda_i) - p^\pi(\lambda_i), \underline{u}_i(\lambda_i, \pi) = \underline{v}_i(\lambda_i) - p^\pi(\lambda_i), \tilde{u}_i(\lambda_i, \pi) = \tilde{v}_i(\lambda_i) - p^\pi(\lambda_i)$, where $\tilde{v}_i$ is defined with respect to candidate passing trade $\breve{\lambda}_i$. The high-level approach is as follows:

function UpdateMRPAR
  $\breve{\lambda}_i \in \arg\max_{\lambda_i \in \mathcal{F}_i(x^0)} u_i(\lambda_i, \pi)$
  if $u_i(\breve{\lambda}_i, \pi) < 0$ **then**
    reduce slack on $\breve{\lambda}_i$ by $u_i(\breve{\lambda}_i, \pi)$
  end if
  $\lambda_i^U \in \arg\max_{\lambda_i \in F_i(x^0)} \tilde{u}_i(\lambda_i, \pi)$
  while $\underline{u}_i(\breve{\lambda}_i, \pi) < \tilde{u}_i(\lambda_i^U, \pi)$ **do**
    Heuristically reduce upper bounds on $\lambda_i^U \setminus \breve{\lambda}_i$ by $\tilde{u}_i(\lambda_i^U, \pi) - \underline{u}_i(\breve{\lambda}_i, \pi)$
    If remaining slack heuristically reduce lower bounds on $\breve{\lambda}_i \setminus \lambda_i^U$
    $\lambda_i^U \in \arg\max_{\lambda_i \in \mathcal{F}_i(x^0)} \tilde{u}_i(\lambda_i, \pi)$
  end while
  if $\breve{\lambda}_i \neq \lambda_i^\alpha$ **then**
    while $\underline{u}_i(\breve{\lambda}_i, \pi) \leq \tilde{u}_i(\lambda_i^\alpha, \pi)$ **do**
      Heuristically reduce upper bounds on $\lambda_i^\alpha \setminus \breve{\lambda}_i$ by $\tilde{u}_i(\lambda_i^\alpha, \pi) - \underline{u}_i(\breve{\lambda}_i, \pi)$
      If remaining slack heuristically reduce lower bounds on $\breve{\lambda}_i \setminus \lambda_i^\alpha$

end while
end if
return $\breve{\lambda}_i$
end function

The bidding agent makes use of a couple of optimization modalities that are exposed by the proxy to the bidder. The procedure first chooses the most preferred trade at truth as the trade to pass MRPAR with $\breve{\lambda}_i$; the bidding agent requests that the proxy finds this trade by solving a MIP. If the trade has negative profit, then the bidding agent attempts to demonstrate positive profit for this trade. Next, the bidding agent enters a loop, wherein it repeatedly requests the proxy to run a MIP that calculates a witness trade λ_i^U with respect to $\breve{\lambda}_i$. As long as this witness has more profit than that of what should be the most preferred trade, the bidding agent adjust bounds so as to reverse this mis-ordering. Lastly, because the bidding agent must pass MRPAR, not merely RPAR, the bidding agent attempts to show a strict preference for $\breve{\lambda}_i$ over λ_i^α when they are not identical.

In meeting DIAR, the bidding agent responds to the $\Delta^F \geq 0$ and $\epsilon \geq 0$ parameter provided by the proxy as follows. Let λ^F be the trade chosen in the maximization that calculates Δ^F. The high-level approach is as follows:

function UPDATEDIAR
while Proxy says we still have not passed DIAR **do**
if λ^F or λ^α can be modified to reduce DIAR error by ϵ over last round **then**
Heuristically reduce the upper-bound slack in $\lambda^F \setminus \lambda^\alpha$
Heuristically reduce the lower-bound slack in $\lambda^\alpha \setminus \lambda^F$
else
Heuristically reduce the upper-bound slack in $\lambda^\alpha \setminus \lambda^F$
Heuristically reduce the lower-bound slack in $\lambda^F \setminus \lambda^\alpha$
end if
end while
end function

The bidding agent attempts to make the current failing trade pass DIAR if possible by reducing the error with respect to that trade. Otherwise, it reduces bounds to prove that DIAR could not be made to pass on that trade and loops on to the next trade.

References

Ausubel, L., Cramton, P., & Milgrom, P. (2006). The clock-proxy auction: A practical combinatorial auction design. In Cramton et al. (Cramton et al., 2006), chap. 5.

Ausubel, L. M., & Milgrom, P. (2002). Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, *1*, 1–42.

Babaioff, M., & Walsh, W. E. (2005). Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation. *Decision Support Systems*, *39*, 123–149.

Ball, M., Donohue, G., & Hoffman, K. (2006). Auctions for the safe, efficient, and equitable allocation of airspace system resources. In Cramton et al. (Cramton et al., 2006), chap. 20.

Ball, M. O., Ausubel, L. M., Berardino, F., Cramton, P., Donohue, G., Hansen, M., & Hoffman, K. (2007). Market-based alternatives for managing congestion at new york's laguardia airport. In *Proceedings of AirNeth Annual Conference*.

Bererton, C., Gordon, G., & Thrun, S. (2003). Auction mechanism design for multi-robot coordination. In *Proc. 17th Annual Conf. on Neural Information Processing Systems (NIPS'03)*.

Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to Linear Optimization*. Athena Scientific.

Bikhchandani, S., & Mamer, J. W. (1997). Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, *74*, 385–413.

Bikhchandani, S., & Ostroy, J. M. (2002). The package assignment model. *Journal of Economic Theory*, *107*(2), 377–406.

Blum, A., Sandholm, T., & Zinkevich, M. (2006). Online algorithms for market clearing. *Journal of the ACM*, *53*, 845–879.

Boutilier, C. (2002). Solving concisely expressed combinatorial auction problems. In *In Proceedings of the 18th National Conference on Artificial Intelligence*, pp. 359–366.

Boutilier, C., & Hoos, H. (2001). Bidding languages for combinatorial auctions. In *Proc. 17th International Joint Conference on Artificial Intelligence*, pp. 1121–1217.

Bredin, J., Parkes, D. C., & Duong, Q. (2007). Chain: A dynamic double auction framework for matching patient agents. *Journal of Artificial Intelligence Research*, *30*, 133–179.

Cavallo, R., Parkes, D. C., Juda, A. I., Kirsch, A., Kulesza, A., Lahaie, S., Lubin, B., Michael, L., & Shneidman, J. (2005). TBBL: A Tree-Based Bidding Language for Iterative Combinatorial Exchanges. In *Multidisciplinary Workshop on Advances in Preference Handling (IJCAI)*.

Chu, L. Y., & Shen, Z. M. (2007). Truthful double auction mechanisms for e-marketplace. *Operations Research*. To appear.

Compte, O., & Jehiel, P. (2007). Auctions and information acquisition: Sealed-bid or Dynamic Formats?. *Rand Journal of Economics*, *38*(2), 355–372.

Conen, W., & Sandholm, T. (2001). Preference elicitation in combinatorial auctions.. In Wellman, & Shoham (Wellman & Shoham, 2001), pp. 256–259.

Cramton, P. (2003). Electricity Market Design: The Good, the Bad, and the Ugly. In *Proceedings of the Hawaii International Conference on System Sciences*.

Cramton, P. (2006). Simultaneous ascending auctions. In Cramton et al. (Cramton et al., 2006), chap. 3.

Cramton, P., Kwerel, E., & Williams, J. (1998). Efficient relocation of spectrum incumbents. *Journal of Law and Economics*, *41*, 647–675.

Cramton, P., Shoham, Y., & Steinberg, R. (Eds.). (2006). *Combinatorial Auctions*. MIT Press.

Day, R., & Raghavan, S. (2007). Fair payments for efficient allocations in public sector combinatorial auctions. *Management Science*, *53*(9), 1389.

de Vries, S., Schummer, J., & Vohra, R. V. (2007). On ascending Vickrey auctions for heterogeneous objects. *Journal of Economic Theory*, *132*, 95–118.

de Vries, S., & Vohra, R. V. (2003). Combinatorial auctions: A survey. *Informs Journal on Computing*, *15*(3), 284–309.

Dias, M., Zlot, R., Kalra, N., & Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, *94*, 1257–1270.

Dunford, M., Hoffman, K., Menon, D., Sultana, R., & Wilson, T. (2003). Testing linear pricing algorithms for use in ascending combinatorial auctions. Tech. rep., SEOR, George Mason University. Submitted to INFORMS J.Computing.

Fu, Y., Chase, J., Chun, B., Schwab, S., & Vahdat, A. (2003). Sharp: an architecture for secure resource peering. In *Proceedings of the 19th ACM symposium on Operating systems principles*, pp. 133–148. ACM Press.

Gerkey, B. P., & Mataric, M. J. (2002). Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems*, *18*, 758–768.

Hudson, B., & Sandholm, T. (2004). Effectiveness of query types and policies for preference elicitation in combinatorial auctions. In *Proc. 3rd Int. Joint. Conf. on Autonomous Agents and Multi Agent Systems*, pp. 386–393.

Kelso, A. S., & Crawford, V. P. (1982). Job matching, coalition formation, and gross substitutes. *Econometrica*, *50*, 1483–1504.

Krishna, V. (2002). *Auction Theory*. Academic Press.

Kwasnica, A. M., Ledyard, J. O., Porter, D., & DeMartini, C. (2005). A new and improved design for multi-object iterative auctions. *Management Science*, *51*, 419–434.

Kwerel, E., & Williams, J. (2002). A proposal for a rapid transition to market allocation of spectrum. Tech. rep., FCC Office of Plans and Policy.

Lahaie, S., Constantin, F., & Parkes, D. C. (2005). More on the power of demand queries in combinatorial auctions: Learning atomic languages and handling incentives. In *Proc. 19th Int. Joint Conf. on Artificial Intell. (IJCAI'05)*.

Lahaie, S., & Parkes, D. C. (2004). Applying learning algorithms to preference elicitation. In *Proc. 5th ACM Conf. on Electronic Commerce (EC-04)*, pp. 180–188.

McAfee, R. P. (1992). A dominant strategy double auction. *J. of Economic Theory*, *56*, 434–450.

Milgrom, P. (2004). *Putting Auction Theory to Work*. Cambridge University Press.

Milgrom, P. (2007). Package auctions and package exchanges (2004 Fisher-Schultz lecture). *Econometrica*, *75*, 935–966.

Mishra, D., & Parkes, D. C. (2007). Ascending price Vickrey auctions for general valuations. *Journal of Economic Theory*, *132*, 335–366.

Myerson, R. B., & Satterthwaite, M. A. (1983). Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, *28*, 265–281.

Nemhauser, G., & Wolsey, L. (1999). *Integer and Combinatorial Optimization*. Wiley-Interscience.

Nisan, N. (2006). Bidding languages for combinatorial auctions. In Cramton et al. (Cramton et al., 2006), chap. 9.

Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. (Eds.). (2007). *Algorithmic Game Theory*. Cambridge University Press.

O'Neill, R. P., Sotkiewicz, P. M., Hobbs, B. F., Rothkopf, M. H., & Stewart, Jr., W. R. (2005). Efficient market-clearing prices in markets with nonconvexities. *European Journal of Operations Research*, *164*, 269–285.

Parkes, D. C. (2007). On-line mechanisms. In Nisan et al. (Nisan, Roughgarden, Tardos, & Vazirani, 2007), chap. 16. To appear.

Parkes, D. C., Kalagnanam, J. R., & Eso, M. (2001). Achieving budget-balance with Vickrey-based payment schemes in exchanges. In *Proc 17th International Joint Conference on Artificial Intelligence*, pp. 1161–1168.

Parkes, D. C., & Ungar, L. H. (2000a). Iterative combinatorial auctions: Theory and practice. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pp. 74–81.

Parkes, D. C., & Ungar, L. H. (2000b). Preventing strategic manipulation in iterative auctions: Proxy agents and price-adjustment. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pp. 82–89.

Rassenti, S. J., Smith, V. L., & Bulfin, R. L. (1982). A combinatorial mechanism for airport time slot allocation. *Bell Journal of Economics*, *13*, 402–417.

Rothkopf, M. H., Pekeč, A., & Harstad, R. M. (1998). Computationally manageable combinatorial auctions. *Management Science*, *44*(8), 1131–1147.

Saatcioglu, K., Stallaert, J., & Whinston, A. B. (2001). Design of a financial portal. *Communications of the ACM*, *44*, 33–38.

Sandholm, T. (2006). Optimal winner determination algorithms. In Cramton et al. (Cramton et al., 2006), chap. 14.

Sandholm, T. (2007). Expressive Commerce and Its Application to Sourcing: How We Conducted $35 Billion of Generalized Combinatorial Auctions. *AI Magazine*, *28*(3), 45.

Sandholm, T., & Boutilier, C. (2006). Preference elicitation in combinatorial auctions. In Cramton et al. (Cramton et al., 2006), chap. 10.

Shapley, L. S., & Shubik, M. (1972). The assignment game I: The core. *Int. Jounral of Game Theory*, *1*, 111–130.

Smith, T., Sandholm, T., & Simmons, R. (2002). Constructing and clearing combinatorial exchanges using preference elicitation. In *AAAI-02 workshop on Preferences in AI and CP: Symbolic Approaches*.

Vossen, T. W. M., & Ball, M. O. (2006). Slot trading opportunities in collaborative ground delay programs. *Transportation Science*, *40*, 15–28.

Wellman, M. P., & Shoham, Y. (Eds.). (2001). *Proc. 3rd ACM Conf. on Electronic Commerce (EC-01)*, New York, NY. ACM.

Wurman, P. R., & Wellman, M. P. (2000). A*k*BA: A progressive, anonymous-price combinatorial auction. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, pp. 21–29.