Combinatorial Markets in Theory and Practice: Mitigating Incentives and Facilitating Elicitation

A dissertation presented
by

Benjamin Lubin

to
The Department of Computer Science
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of

Computer Science

Harvard University
Cambridge, Massachusetts
November 2010

Advisor

David C. Parkes

Author

Benjamin Lubin

**Combinatorial Markets in Theory and Practice:
Mitigating Incentives and Facilitating Elicitation**

Abstract

Strategyproof mechanisms provide robust equilibria with minimal assumptions about knowledge and rationality, but can be unachievable in combination with other desirable properties, such as budget-balance, stability against deviations by coalitions, and computational tractability. We thus seek a relaxation of this solution concept, and propose several definitions for general settings with private and quasi-linear utility. We are then able to describe the ideal mechanism according to these definitions by formulating the design problem as a constrained optimization problem. Discretization and statistical sampling allow us to reify this problem as a linear program to find ideal mechanisms in simple settings. However, this constructive approach is not scalable.

We thus advocate for using the quantiles of the *ex post* unilateral gain from deviation as a method for capturing useful information about the incentives in a mechanism. Where this also is too expensive, we propose using the KL-Divergence between the payoff distribution at truthful reports and the distribution under a strategyproof "reference" mechanism that solves a problem relaxation. We prove bounds that relate such quasimetrics to our definitions of approximate incentive compatibility; we demonstrate empirically in combinatorial market settings that they are informative about the eventual equilibrium, where simple regret-based metrics are not.

We then design, implement, and analyze a mechanism for just such an overconstrained setting: 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. A proxied interpretation of a revealed-preference activity rule, coupled with simple linear prices, ensures progress across rounds. We are able to prove efficiency under truthful bidding despite using linear pricing that can only approximate competitive equilibrium. Finally, we apply several key concepts from this general mechanism in a combinatorial market for finding the right balance between power and performance in allocating computational resources in a data center.

Contents

Citations to Previously Published Work

A portion of the material presented in Chapter 4 has previously appeared as:

> "Quantifying the Strategyproofness of Mechanisms via Metrics on Payoff Distributions", B. Lubin, and D. C. Parkes, Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI-09), 18-21, 2009.

The majority of the material in Chapter 5 has appeared as referenced below. However, additional details are provided herein, particularly as pertains to the inter-round pricing method.

> "ICE: An Expressive Iterative Combinatorial Exchange", B. Lubin, A. Juda, R. Cavallo, S. Lahaie, J. Shneidman, and D. C. Parkes, Journal of Artificial Intelligence Research (JAIR) **33**, 33-77, 2008.

Early work on this system appeared in:

> "ICE: An Iterative Combinatorial Exchange", D. Parkes, R. Cavallo, N. Elprin, A. Juda, S. Lahaie, B. Lubin, L. Michael, J. Shneidman, H. Sultan, Proceedings of the 6th ACM Conference on Electronic Commerce (EC-05), 249-258, 2005.

A concise version of the material in Chapter 6 appeared as:

> "Expressive Power-Based Resource Allocation for Data Centers", B. Lubin, J. Kephart, R. Das, and D. C. Parkes, Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09), 1451-1456, 2009.

Acknowledgements

First of all, my profoundest thanks to my advisor David Parkes, who has been a constant inspiration and mentor. His absolutely consistent intelligence, energy, and most of all kindness have made my graduate experience a total joy. I am convinced that there are but few others who have had such an unqualifiedly positive experience in pursuing their Ph.D. and I have David to thank for this. A huge thanks also to my committee members, Tuomas Sandholm, Leslie Valiant and Greg Morrisett, for their comments, guidance and encouragement.

My thanks as well to Radhika Nagpal and Yiling Chen for their support durring my graduate studies. Barbara Grosz catalyzed and then cemented my interest in AI. Her mentorship was hugely important in my undergraduate years, and in my deciding to return for graduate studies, for which I am grateful. I am also grateful to John Parsons and Alan Gara for giving me the opportunity to work on their research, and for being both great role models and mentors.

I owe a deep debt to my coauthors on the various papers whose content is reflected in this thesis. My particular thanks go to those who had a hand in shaping the ICE project, including Adam Juda, Ruggiero Cavallo, Sébastien Lahaie, Jeffrey Shneidman, Loizos Michael, Hassan Sultan, and Nick Elprin. Also to Jeff Kephart and Rajarshi Das for a wonderful summer spent at IBM Research Hawthorne, for their hard work on our paper, and for their continuing support. My thanks, as well, to the anonymous reviewers on these papers whose comments and suggestions proved hugely valuable.

To my wife Elisa and my daughter Sophie: my deepest thanks for all of the encouragement, and for tolerating the long hours it has taken to bring this research to fruition. I love you both. Also to my family, Steve, Wendy, Nathaniel, David and Margaret – this document could not have happened without your encouragement and love. Finally, to my friends both at the graduate school and beyond it: thank you.

I am grateful to the following institutions for supporting my graduate studies: BBN Technologies, Yahoo! Research, IBM Research, and the Siebel Foundation for their very generous fellowship.

Dedicated to my parents Steve and Wendy,
my brother Nathaniel,
my wife Elisa,
and my daughter Sophia.

Also to
Sally Aberth and Toni Gould,
who opened this path to me.

The real price of everything, what everything really costs to the man who wants to acquire it, is the toil and trouble of acquiring it.

– Adam Smith
The Wealth of Nations

1

Introduction

In 2002, Google transformed sponsored search by implementing a multi-item second-price auction, outclassing earlier first-price auctions that had been pioneered by GoTo.com (later Overture/Yahoo!) [Jansen and Mullen, 2008]. This innovation vastly improved the efficiency of the search engine business model, generating both greater value for the advertiser and increased profits for Google. In the old model, advertisers had strong monetary incentives to deviate from bidding their true value, leading to social welfare losses and the need for constant, unending strategic tinkering of bid amounts to respond to the actions of other advertisers. Google's mechanism design innovation dramatically streamlined this market, taking to heart the message underlying the Adam Smith quotation above, that the success of markets is in large measure determined by the ease and efficiency of the transaction itself.

Google's improvement is one example of how mechanism design addresses the problem of achieving desirable outcomes in multi-agent systems despite the anti-optimizing effects of private information about valuations and individual self-interest. Mechanism design finds applications in societal contexts (e.g., school and medical-residents matching [Abdulkadiroglu, Pathak, and Roth, 2005]) and business contexts (e.g., sponsored search auctions [Lahaie, Pennock, Saberi, and Vohra, 2006]), while providing a formal paradigm by which to coordinate the behavior of artificial agents (e.g., for task and resource allocation) [Bererton, Gordon, and Thrun, 2003; Dias, Zlot, Kalra, and Stentz, 2006; Gerkey and Mataric, 2002; Sandholm, 1993; Zlotkin and Rosenschein, 1996].

In designing mechanisms, there are a number of properties that we would like to achieve, such as:

Efficiency: The chosen outcome should maximize the total value for participants.

Incentive Compatibility: In equilibrium, agents should choose to behave *truthfully*.

Individual Rationality: Agent participation should be voluntary.

Budget Balance: Running the mechanism should not require a subsidy.

Core Outcome: Coalitions of participants should not wish to reject the chosen outcome in favor of their own.

However, theory has shown that a simultaneous realization of even the first four of these properties is not achievable [e.g. Myerson and Satterthwaite, 1983]. We argue that properties such as *budget balance* and *individual rationality* should be taken as fixed, as mechanisms will often be rejected without them. In some contexts it is desirable to add the *Core Outcome* requirement to this list as well. Choosing the exact set of constraints necessary for a given setting is not our purpose here. Rather, we address the question of how to handle the over-constrained nature of the design problem overall. Clearly some relaxation is necessary. We argue that, given the choice, it is generally appropriate to relax the requirements for *incentive compatibility* and, thus, *efficiency*. Note that a mechanism that fails to induce truthful behavior in its participants cannot be efficient, because it will not have the information necessary to make welfare-maximizing decisions. This raises the question: How can we design mechanisms that are as *incentive-compatible* as possible? This is one of the big open questions in mechanism design, and in this thesis we offer progress on it along several avenues.

1.1 Overview

1.1.1 Approximately Incentive-Compatible Payment Rules

In this thesis we generally restrict ourselves to settings where agents have *quasi-linear* preferences, meaning that their *utility* is the difference between the value they assign to a given outcome, and the price they pay to achieve that outcome (e.g., where the outcome is an allocation of resources). We then ask the question: What *payment rule* should the mechanism adopt that will best incentivize agents to be truthful in their interactions with the mechanism? And just as importantly: What precisely do we mean by "best"?

The gold standard for approximate incentive compatibility, as well as approximate efficiency, is behavior in *Bayes-Nash Equilibrium* (BNE). In BNE, each agent picks a *strategy* for playing the game induced by the mechanism that is expected-case

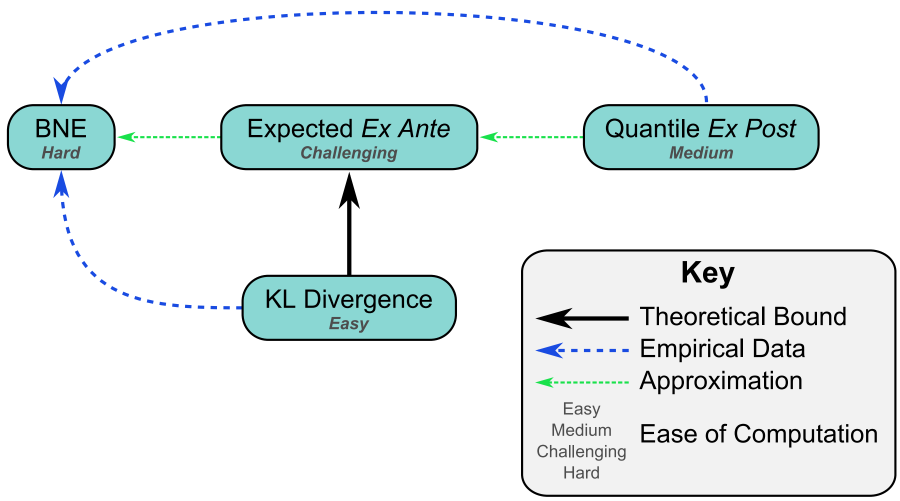


Figure 1.1: Overview of work in Chapters 3 and 4 on characterizing approximately incentive compatible payment rules.

maximal over a commonly known joint distribution of other agents' values (known as *types*). However, the BNE of a mechanism is extremely difficult to calculate computationally. We thus are interested in cheaper indirect measures of approximate incentive compatibility, that will correlate well with behavior in equilibrium. In Chapter 3, we first define several of these based on unilateral incentives for deviation. We then examine a constructive approach for designing rules that best meet these targets. While this is instructive about the nature of optimal payment rules at a theoretical level, and in fact is practically achievable for small problem sizes, it will not scale to large, complex real-world problems.

We therefore switch to an analytic approach that seeks to determine which rule best achieves our goals from among a set of predefined, closed-form payment rules. The question then becomes how to perform this analysis in a way that is tractable for the complex settings we are interested in (since calculating a full BNE is so difficult).

Figure 1.1 summarizes this work on measuring approximate incentive compatibility. At the top left, we have the BNE outcome. In our experimental work, we go to considerable lengths to achieve approximations for the equilibrium play of the mechanisms being evaluated in order to properly evaluate the design criteria we are proposing against the gold standard. In order to do this, we limit our experiments to problems that, while large enough to be interesting, are sufficiently small to remain tractable.

We believe that a particular measure of the bidding behavior that we propose, the expected *ex ante* unilateral incentive to deviate, should be a very good indication of equilibrium behavior. However, this too is challenging to compute because a stochastic optimization must be done so as to limit the model of agent information away from full information and to a distribution of the joint value space.

Instead we propose looking at the quantiles of the distribution of *ex post* unilateral incentive to deviate. This measure still requires an optimization over agent behavior, but it is a full information optimization that can often exploit the structure of the mechanism to become a simple linear or grid search.

While this *ex post* quantile approach is very appealing, we recognize that in some domains even this option may be overly complex. So we also propose using a simple KL-Divergence to measure the distance between the distribution of profits under the approximately incentive-compatible mechanism being evaluated, and some reference mechanism that offers no incentives for lying, but which fails some other condition that makes it unusable directly. We show experimentally that it correlates well with behavior in the BNE, and offer a proof that this measure will bound the *ex ante* unilateral deviation measure.

1.1.2 The Construction of Combinatorial Markets

In domains where there are incumbents with property rights, it is necessary to facilitate a complex multi-way reallocation of resources. Combinatorial exchanges (CE) [Parkes, Kalagnanam, and Eso, 2001a] solve this problem by combining and generalizing two different mechanisms: double auctions and combinatorial auctions. In a double auction (DA), multiple buyers and sellers trade units of an identical good [McAfee, 1992b]. In a combinatorial auction (CA), a single seller has multiple heterogeneous items up for sale [Cramton, Shoham, and Steinberg, 2006; de Vries and Vohra, 2003]. 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. But it is known that one cannot simultaneously maximize welfare while retaining the desirable properties of *individual rationality*, budget balance and incentive compatibility, as described above.

A CE is thus an ideal environment in which to situate the payment rules we have been examining. In addition to adopting CE's as a domain to study the design of payment rules, we conduct an extensive study of the design of an *iterative* CE. This is primarily focused on the orthogonal question of how to elicit preferences efficiently. The mechanism we propose, the *Iterative Combinatorial Exchange* (ICE), is general, and has many other desirable properties.

Calculating the value of any individual bundle of goods that a given agent may acquire in the market may necessitate the solution of a difficult optimization problem (e.g. vehicle routing, scheduling, etc). Thus we want to permit agents to reveal only incremental information to the market, which they refine through rounds. The process is analogous to how the price and allocation are slowly determined through rounds in the standard single-item English auction used in public auction houses.

In order to implement this *iterative* process, however, we need a set of tools that must seamlessly work in concert to produce an overall design that is computationally tractable, has the economic properties that we require, and does not place undue computational burdens on its participants.

First, agents require a way to bid in the system, and as their valuations are highly complex, we need a bidding language that can capture the values of complex combinations of bundles concisely. For this we propose the *TBBL* language which, as we shall see, has extremely intuitive semantics. Moreover, as most agents will desire to reveal information incrementally, the *TBBL* language permits agents to specify only upper and lower bounds on their valuations which they then refine over time.

Second, we need a way to provide prices that agents can use to guide this information revelation process. In the complex economy of a CE, we would need non-anonymous non-linear prices to achieve a full competitive equilibrium. However, as non-anonymous bundle prices are extremely hard for agents to reason about, our design quotes linear pricing during the elicitation process, and only uses non-linear non-anonymous *payments*–whose game theoretic properties we have been describing above–at the very end, to define the terminal payments that agents actually pay. We choose these prices extremely carefully, to ensure that they are unique, are as close to the competitive equilibrium as possible, and that they imply these terminal payments as closely as possible.

Third, we need to specify activity rules which force agents to provide information to the system. Without such rules, an agent would choose to remain silent, letting other agents' bidding activity provide pricing information which could then be optimally exploited in the last round before the exchange closes. In response, we propose the *Modified Revealed Preference Activity* rule (loosely based on the work of Ausubel, Cramton, and Milgrom [2006]), which forces agents to make clear which trade they most prefer at the current prices, and thus moves the allocation towards the efficient outcome if the prices are accurate enough. As our linearized prices may not be perfectly accurate, we require an additional activity rule, the *Delta Improvement Activity Rule*, to drive additional elicitation (though in practice our prices are typically good enough to require only minimal use of the DIAR rule).

We evaluate ICE extensively, and are able to show the following: that it is scalable to practical problem sizes; that its iterative nature does not produce undue computational burdens on participants; that the economic properties of its inter-round prices are such that they provide useful feedback to bidders; and that the mechanism when taken as a whole can quickly find the efficient choice of outcome without requiring too many rounds or too much information revelation. The design is agnostic as to the rule used to determine the final payments, and thus it can be used in concert with the innovations proposed in the rest of the thesis.

Finally, we demonstrate the effectiveness of combinatorial markets in an important specific setting: power-aware resource allocation in data centers. We show that a carefully constructed market in this setting can enable an economically motivated balance between the power being consumed and the performance (and thus utility) being generated by the data center. Additionally, the market enables the data center operator to be principled about how he rations limited resources when the demand of the various clients outstrips delivery capacity. We show that the market can improve the quality of resource and power allocation in realistic scenarios, and that a market that can be operated on a single machine can allocate the resources of 1000 others, and thus does not induce so large an overhead as to cancel out the additional power efficiency it creates.

1.2 Key Technical Contributions

The most important technical contributions of the work in this thesis include:

- Introduction of several *indirect* criteria for approximate incentive compatibility that we argue are appropriate targets for design including the expected *ex ante* and quantile *ex post* unilateral gain by deviating from truthful reporting.
- Formulation of the ideal mechanism according to these criteria as a constrained optimization problem, and presentation of tractable LP approximations to this problem via statistical sampling.
- Calculation of partially symmetric BNE approximations in a complex CE environment.
- Identification of the KL-Divergence between the distribution of payoffs in a given mechanism and the distribution of payoffs in a strategyproof reference as a computationally simple way to predict behavior in the BNE.

- Provision of a theorem that interprets this metric in terms of the earlier criterion for approximate incentive compatibility.
- Evaluation of the *ex post quartile* criterion as a way to capture rich information about approximate incentive compatibility, that while more computationally expensive than the KL-Divergence approach, is much more tractable than a full BNE calculation.
- Definition of unique, linear and approximately competitive equilibrium prices in a setting that would normally require non-linear, non-anonymous prices; also, a highly sophisticated method for computing these prices using heuristically guided LP search with optimized constraint generation.
- Furnishing of two activity rules that together force information revelation in the mechanism, and thus guarantee termination with the efficient allocation (at reports) without requiring the agents to calculate or reveal their full valuation profile, and despite using only linear prices.
- Description of a concrete combinatorial market that includes a sophisticated model of the power usage of modern data center hardware to high fidelity.
- Creation of a bidding proxy for use in such a market that predicts buyer demand, models application performance given supply and demand, and handles conversion from long run *Service Level Agreements* to a short term spot market.

1.3 Domains for Combinatorial Markets

The types of complex combinatorial mechanisms in which we are interested in this thesis are applicable to a wide range of settings. By way of motivation, we list several of these:

Financial Markets:

There are many potential applications of CAs or CEs to financial markets. Firms interested in certain hedging transactions or in portfolio re-balancing often want to execute complex combinations of trades in an all-or-nothing way [Saatcioglu, Stallaert, and Whinston, 2001], as buying or selling these instruments individually opens traders to either undesirable price movements or risk of exposure. Some brokers presently allow for bundled “multi-leg” options trades. Such trades can be cleared using a combinatorial mechanism without the execution risk inherent in simply disaggregating the bundles, even when the broker has large volume and fast trades.

Multi-Agent Task Allocation:

Combinatorial markets can be used for distributed task allocation in face of private interest and private information. They can therefore be a powerful tool for coordination in multi-agent systems [Bererton et al., 2003; Dias et al., 2006; Gerkey and Mataric, 2002; Sandholm, 1993; Zlotkin and Rosenschein, 1996].

Airport Slot Auctions:

Some of the earliest work on combinatorial mechanisms was motivated by the need to allocate landing and takeoff rights at highly congested airports (such as JFK, LaGuardia, O'Hare and National) [Ball, Ausubel, Berardino, Cramton, Donohue, Hansen, and Hoffman, 2007; Rassenti, Smith, and Bulfin, 1982]. The airport domain is exceedingly complex. It involves multiple time scales (long-term strategic assignments, and short-term reassignments, e.g. under weather-induced congestion). Participants have existing property rights, typically requiring an exchange, not an auction. Agents have both powerful complements and substitutes in their valuation functions – and calculating the value of even a single assignment profile may require execution of a computationally expensive vehicle routing problem [Ball, Donohue, and Hoffman, 2006]. Recent work has focused on improvements to the structure of the short term "tactical" reassignment problem [Vossen and Ball, 2006].

Bandwidth Auctions:

Some of the most public successes of mechanism design have been in the government auction of wireless spectrum [Kwerel and Williams, 2002]. These designs have carefully taken into account the complements that agents typically have over blocks of spectrum [Milgrom, 2004]. Careful design is needed to balance the desire for efficiency with typical requirements for reasonable government revenue [Day and Cramton, 2008]. The domain is made even more difficult when existing property owners must be moved off their holdings (e.g. existing analog TV stations; see [Cramton, Kwerel, and Williams, 1998]); the need to handle such reallocation problems was the motivation for some of the work in Chapter 5.

Advertising Auctions:

Another highly visible application of mechanism design is to markets for *sponsored search* advertising at firms like Google, Yahoo!, and Microsoft. These firms have settled on the *Generalized Second Price* auction mechanism. This mechanism allocates the available slots in order of decreasing bid size, and then charges a price for each slot that is a function of the bid for the next lowest slot. This mechanism has the advantages of being simple, computationally tractable, and high in seller revenue, which has led to its use over the theoretically compelling VCG mechanism [Edelman, Ostrovsky, and Schwarz, 2007]. The framework for design that we propose in Chapter 3 is ostensibly applicable to this setting.

The market for the sale of display advertising on various web pages is also huge and growing (it was worth $7.7 billion in 2008 [Evans, 2009]). Most of the premium placements (that comprise much of the dollar-weighted value in the market) are still manually negotiated, producing inefficient outcomes. To clear this market optimally, a multi-party matching problem must be solved between the advertisers and the publishers. Consequently, AdECN (Microsoft), RightMedia (Yahoo!), and DoubleClick (Google) have each implemented new exchanges that permit adverts obtained from *advertising networks* (effectively brokers acting on behalf of sets of participants from a given side of the market) to be placed on specific web sites [Muthukrishnan, 2009]. As these exchanges become more expressive, and with more forward planning with contracts and commitments to particular quantities of impression types to particular advertisers, these markets could benefit from versions of the CE design proposed in Chapter 5.

Supply Chain Auctions:

CEs have promise as mechanisms for expressive sourcing by multiple bid-takers, perhaps representing different profit centers within an organization, or across organizations. CombineNet has operated combinatorial auctions for supply chain management and procurement that have sold more than $35 billion worth of goods and services for major companies such as Walmart, Target and Proctor & Gamble [Sandholm, 2007]. These auctions have saved the companies that used them at least $4 billion relative to what they would have spent for the same items at the prices they had obtained under their earlier procurement procedures. Clearly, a better mechanism can generate a tremendous amount of savings, and just as importantly, increase the efficiency of allocations. This incredibly impressive performance is perhaps the best example of the actual commercial use of combinatorial market technologies to date.

Computational Resource Allocation:

There has been considerable interest in using market techniques to allocate computing resources either in data centers [e.g. Byde, Salle, and Bartolini, 2003; Ferguson, Nikolaou, Sairamesh, and Yemini, 1996; Preist, Byde, Bartolini, and Piccinelli, 2002], or in distributed systems [e.g. Fu, Chase, Chun, Schwab, and Vahdat, 2003; Regev and Nisan, 2000; Waldspurger, Hogg, Huberman, Kephart, and Stornetta, 1992]. As data centers have become ever larger and more power-hungry, the need to intelligently allocate power has grown as well [Chase, Anderson, Thakar, Vahdat, and Doyle, 2001]. In Chapter 6, we investigate the use of combinatorial mechanisms for this problem in particular.

1.4 Outline

In Chapter 2, we provide classic results from the fields of mechanism design and microeconomics that set the stage for the work discussed in the subsequent chapters. Chapter 3 proposes several *indirect* criteria for measuring approximate incentive compatibility, and then shows how to formulate the problem of how to construct an optimal payment rule under these criteria as a constrained optimization problem. In Chapter 4, we then construct a near-BNE for a complex CE domain, and use this to evaluate an even simpler measure for approximate incentive compatibility based on the KL-divergence of payoffs within a given mechanism and a strategyproof reference. We relate this distributional approach to the previously defined criteria, and show the quantile *ex post* condition to be particularly attractive for design. Chapter 5 provides a complete design, implementation and evaluation of an iterative CE, including a novel bidding language, pricing structure for bidder feedback, and activity rules. Chapter 6 provides a case study in the application of these techniques to the particular problem of power-aware resource allocation in corporate data centers. Finally, in Chapter 7 we offer conclusions and suggestions for future work.

Knowledge, in truth, is the great sun in the firmament. Life and power are scattered with all its beams.

– Daniel Webster
Address at Bunker Hill Monument, 1825

2

Background Material

In this chapter we introduce a number of important definitions, concepts and mathematical constructions that we will need in later chapters. In the course of these descriptions, we survey some of the most important classical results from game theory and mechanism design.

2.1 Mechanism Design

Mechanism design is the study of how to construct procedures in a setting involving multiple self-interested participants, such that when the procedures get activated by specific inputs, the participants choose socially desirable outcomes. Important examples where money is permitted to be exchanged include auctions, stock exchanges, and supply-chain sourcing. Recent work has also focused on settings where monetary exchange is prohibited, including such diverse examples as municipal school choice, kidney exchanges and university course assignments. There is an extensive and ever-growing literature on mechanism design, but a good overview of recent advances, with a focus on the area of auction theory with which we are concerned, is to be found in Milgrom [2004]. The following discussion closely follows that in Parkes [2001] in providing definitions and basic results in mechanism design.

A mechanism's task is to determine a particular outcome o from a set of possibilities $\mathcal{O}$. Each of N participants (or agents) in the mechanism is said to have a specific *type* $\theta_i \in \Theta_i$ which determines his preference.

2.1.1 The Domain of Agent Preferences

The types of preferences that agents may have has a huge impact on the structure and attributes of a good design. In this thesis we will generally be concerned with design settings where agent preferences belong to a domain with the following properties:

Private Value Agents have *private value preferences* if their value for an outcome is independent of the types of the other agents. This stands in contrast to *interdependent value preferences*, where agents' value can depend on the types of the other agents. When agent preferences are private, then they can be expressed in the form of a function $u_i(o, \theta_i) : \mathcal{O} \times \Theta_i \to \mathbb{R}$, which specifies the utility that agent i with type θ_i obtains when the outcome o is chosen by the mechanism.

Quasi-Linearity Agent preferences are said to be *quasi-linear* if the agent types specify a utility function that is separable into a *valuation function* v_i and a payment p_i:

$$u_i(o, \theta_i) = v_i(\kappa, \theta_i) - p_i \tag{2.1}$$

where outcome o specifies a result κ from a *choice set* $\mathcal{K}$ and a payment vector $\boldsymbol{p} \in \mathbb{R}^N$ which expresses a monetary transfer for each agent. Here, the agent's type θ_i determines the valuation function v_i. As an example, in an auction problem, the $\mathcal{K}$ are possible allocations of goods, and the $\boldsymbol{p}$ vector specifies payments to the auctioneer. While this value structure is commonly assumed and is reasonable in many contexts, it does have its limitations. Specifically, it assumes that agents are both risk neutral and not budget constrained.

2.1.2 Defining a Mechanism

A *strategy* specifies what an agent with a given type will do in every possible state of the game. Concretely, if the mechanism defines a game in which agents must choose an *action* in various contexts, then a strategy is a function from the context and type to an action chosen by the agent. We say that agent i adopts strategy $s_i(\theta_i) \in \Sigma_i$ if this choice is deterministic, or *mixed* strategy $\sigma_i(\theta_i)$ if the choice is randomized (i.e. σ_i is a probability distribution over elements of Σ_i). In this chapter we restrict ourselves to deterministic strategies for expository purposes; the results can in general be extended to mixed strategy settings.

Every participant in the mechanism has his own type. The goal in mechanism design is to achieve some particular outcome for every possible joint type profile. Formally, we say the goal is to achieve a particular *social choice function*:

Definition 2.1: *A* ***Social Choice Function*** $f : \Theta_1 \times \ldots \times \Theta_N \to \mathcal{O}$ *specifies an outcome* $f(\boldsymbol{\theta}) \in \mathcal{O}$ *given agent types* $\boldsymbol{\theta}$*, where* $\boldsymbol{\theta} = (\theta_1, \ldots \theta_N)$.

The mechanism design problem is to design a game where the outcome will be $f(\boldsymbol{\theta})$ when the agents have types $\boldsymbol{\theta}$ despite agents having private information and being self interested. Formally we have:

Definition 2.2: *A* ***mechanism*** $m = (\boldsymbol{\Sigma}, g)$ *defines the strategies available to the players* $\boldsymbol{\Sigma} = (\Sigma_1, \ldots \Sigma_N)$ *and an outcome rule* $g : \Sigma_1 \times \ldots \times \Sigma_N \to \mathcal{O}$*, which chooses the outcome* $g(\boldsymbol{s}(\boldsymbol{\theta}))$ *that occurs when the participants execute the strategy profile* $\boldsymbol{s} = (s_1, \ldots s_N)$*.*

Informally, this means that the mechanism specifies the rules that constrain what actions and thus what strategies are available, and then determines what outcome occurs when agents behave as they will. The question is, though, how will the agents actually behave so we can know if our desired outcome $f(\boldsymbol{\theta})$ will actually occur? For this we turn to game theory.

2.2 Solution Concepts

Mechanism design is situated within the larger framework of game theory [Fudenberg and Tirole, 1991; Nash, 1950], within which, *solution concepts* constitute some of the fundamental machinery. Game Theory offers several different flavors of these formal models of what outcome to expect in games with self-interested players. Here we present three of the most common. To do this formally, we need a way to express the value of a particular game structure to a given agent. Specifically, an agent can be said to have a utility for the game as a whole, which is a function of the joint types and strategies of all of the participants: $u_i(\boldsymbol{s}, \boldsymbol{\theta})$, where $\boldsymbol{s}$ is the strategy profile of all of the participants and $\boldsymbol{\theta}$ is the type profile of all of the agents.

2.2.1 Dominant Strategy Equilibrium

In a *dominant strategy* equilibrium, the strategy s_i that every participant chooses has the following property:

$$u_i(\langle s_i; \boldsymbol{s}_{\text{-}i}\rangle, \boldsymbol{\theta}) \geq u_i(\langle s'_i; \boldsymbol{s}_{\text{-}i}\rangle, \boldsymbol{\theta}) \;\forall\; s'_i \neq s_i, \;\forall\; \boldsymbol{s}_{\text{-}i} \in \Sigma_{\text{-}i}, \;\forall\; \boldsymbol{\theta} \tag{2.2}$$

where $\boldsymbol{s}_{\text{-}i}$ denotes the strategies of all the participants but i, and the semicolon syntax denotes vector composition. Informally, this says that an agent of type θ_i has a dominant strategy s_i if the agent prefers s_i to all other strategies regardless of what strategies the other participants choose.

Dominant strategy equilibria are extremely desirable in mechanism design because agents can determine their desired strategy without an assumption of rationality on the part of other players, or in fact without any information about what they will do. However, despite being attractive for these reasons, it is not attainable in many important settings.

2.2.2 Bayes-Nash Equilibrium

With dominant strategy equilibrium often unavailable, the concept of a *Bayes-Nash* equilibrium (BNE) is very useful. A strategy profile $\boldsymbol{s}$ is in BNE if for every agent we have:

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta}_{\text{-}i}}\left[u_i(\langle s_i(\theta_i); \boldsymbol{s}_{\text{-}i}(\boldsymbol{\theta}_{\text{-}i})\rangle, \boldsymbol{\theta})\right] \geq \mathop{\mathbb{E}}_{\boldsymbol{\theta}_{\text{-}i}}\left[u_i(\langle s'_i(\theta_i); \boldsymbol{s}_{\text{-}i}(\boldsymbol{\theta}_{\text{-}i})\rangle, \boldsymbol{\theta})\right] \;\forall\; s'_i \neq s_i, \;\forall\; \theta_i \tag{2.3}$$

where the expectation over types is based on a common *joint prior* on the distribution of agent types. Informally in a BNE, every agent chooses a strategy s_i that maximizes his expected utility according to the joint prior when all of the other agents are simultaneously maximizing their own strategies with respect to the joint prior. Note that since the agent is forming a best-response to the distribution over other agent strategies, he may not actually be best-responding to any individual profile of strategies.

This solution concept is more easily attained than dominant strategy equilibrium. However, it still suffers from multiple equilibria, the assumption of the common prior, and from participants having to assume that other players are responding rationally with respect to it. This last assumption is perhaps the most severe: participants can be irrational for a variety of reasons not the least of which is simply making mistakes.

2.3 Implementation

Recall that the goal of *mechanism design* is to construct a mechanism that achieves a particular *social choice function* f, i.e. causes a particular outcome for each possible profile of agent types.

With definitions of both *mechanisms* (games with a particular structure), and several *solution-concepts* (ways to characterize the kinds of strategies rational expected-utility-maximizing agents would adopt within a given game) we are now in a position to formalize what it means to achieve a particular *mechanism design* goal:

Definition 2.3: *Mechanism* $m = (\mathbf{\Sigma}, g)$ ***implements*** *social choice function* f *if*

$$f(\boldsymbol{\theta}) = g(\boldsymbol{s}^*(\boldsymbol{\theta}))\ \forall\ \boldsymbol{\theta} \in \boldsymbol{\Theta}, \text{ where } \boldsymbol{s}^* \text{ is the equilibrium solution to } m.$$

Informally, a mechanism *implements* a given social choice function if the outcome stipulated by the social choice function occurs within the mechanism when the participants are playing in an equilibrium of the mechanism. Any of the two equilibrium concepts can be chosen for the purposes of the definition; e.g. we might say "mechanism m implements social choice function f in Bayes-Nash equilibrium".

2.4 Incentive Compatibility

Next we consider the conditions under which a mechanism provides incentives to participants to interact in a truthful manner. For this, we first need a few definitions: A *direct revelation* mechanism asks each agent to make a claim about his preferences explicitly. Formally:

Definition 2.4: *A* ***direct revelation*** *mechanism restricts the space of strategies for each agent Σ_i to be exactly the space of possible agent types Θ_i. Accordingly, it has an outcome rule $g : \boldsymbol{\Theta} \to \mathcal{O}$ which chooses an outcome o from reports $\hat{\boldsymbol{\theta}}$. Note that in a quasi-linear environment, g will be composed of a choice rule k and a payment rule p.*

In the equilibrium strategy profile of an *incentive-compatible* direct revelation mechanism, all agents report their true types to the mechanism without deviation. Formally,

Definition 2.5: *An* ***incentive-compatible*** *direct revelation mechanism is one in* *which the equilibrium strategy is $s_i^* = I$, i.e. the identity function.*

Incentive compatibility is always defined with respect to a particular solution concept. For dominant strategy equilibrium we have:

Definition 2.6: *A direct revelation mechanism is **Dominant Strategy Incentive-Compatible**, also called **Strategyproof**, if revealing the truth is the dominant strategy equilibrium.*

For a mechanism involving agents with quasi-linear preferences, *strategyproofness* implies that it is a dominant strategy for agents to report their true valuation:

$$v_i(g(v_i, v_{-i})) - p_i(v_i, v_{-i}) \geq v_i(g(\hat{v}_i, v_{-i})) - p_i(\hat{v}_i, v_{-i}) \; \forall \; v_i, \; \forall \; \hat{v}_i, \; \forall \; v_{-i} \tag{2.4}$$

Participants in strategyproof mechanisms needn't reason about the types or strategies of other agents at all. If they know the mechanism is strategyproof, they immediately know that simply stating their value is the optimal behavior, and can spend all of their time worrying about determining their own value, instead of speculating about the behavior of their competitors.

For a Bayes-Nash equilibrium solution concept we have:

Definition 2.7: *A direct revelation mechanism is **Bayes-Nash Incentive Compatible** if all agents choosing to reveal the truth is a Bayes-Nash equilibrium.*

This is not as powerful a mechanism property, due to the weaker solution concept, but it is easier to achieve.

2.4.1 The Revelation Principle

One of the key results in mechanism design is the *Revelation Principle*, which states:

Theorem 2.1: *[Gibbard, 1973; Green and Laffont, 1977; Myerson, 1979; 1981] Suppose there exists a mechanism m that implements the social choice function f in dominant strategies. Then there is also a direct strategyproof mechanism m' which also implements f.*

There is also a Bayes-Nash solution concept version of the revelation principle:

Theorem 2.2: *Suppose there exists a mechanism m that implements the social choice function f in a Bayes-Nash equilibrium. Then there is a direct Bayes-Nash incentive-compatible mechanism m' which also implements f, if the joint distribution of the agent types is common knowledge to the center as well as to the agents.*

The intuition here is that given the true type of an agent, we can simulate the strategy that a given agent would optimally take when playing in mechanism m. We can thus create a new mechanism m' that asks each agent for his truthful report, and then m' runs this simulation of the optimal agent strategies against the original mechanism m and implements the resulting outcome. Given the optimality of the simulator and the choice of outcome rule, it will be in the participants' best interest to report truthfully to m'. We leave the formal proof to the above sources.

The revelation principle implies that if a mechanism is implementable in a non-direct revelation incentive-compatible mechanism, then it is also implementable in a direct mechanism as well. However, the direct version produced by the above construction may be computationally intractable. Thus, indirect mechanisms remain highly relevant in practice [Conitzer and Sandholm, 2004a; Othman and Sandholm, 2009]. The contrapositive of the revelation principle is extremely useful: If a social choice function is not implementable in a direct revelation incentive-compatible mechanism, then it isn't implementable in a non-direct revelation incentive-compatible mechanism either. This enables the construction of powerful impossibility results, such the one we consider in section 2.8.

2.5 Design Properties

2.5.1 Efficient Choice Functions

We highlight one of the most important choice functions for quasi-linear preferences: **Welfare Maximization**. Choice function $k(\boldsymbol{\theta})$ *maximizes welfare* (or is *efficient*) when:

$$\sum_{i=1}^{N} v_i(k(\boldsymbol{\theta}), \theta_i) \geq \sum_{i=1}^{N} v_i(k'(\boldsymbol{\theta}), \theta_i) \; \forall \, k' : \Theta- > \mathcal{K}, \; \forall \, \boldsymbol{\theta} \tag{2.5}$$

Such a *choice* maximizes the total value over all agents. Note that κ will be coupled with a particular payment rule p to form a social choice function $f(\boldsymbol{\theta}) \equiv (\kappa(\boldsymbol{\theta}), p(\boldsymbol{\theta}))$. When κ is welfare maximizing, then we say that f is welfare maximizing as well.

2.5.2 Mechanism Properties

Given a mechanism operating under a particular solution concept, we can examine whether the mechanism exhibits any of several desirable properties:

Efficiency A mechanism is *efficient* if the implemented social choice function is welfare maximizing.

Budget Balance A mechanism is *ex ante budget balanced* if the net transfers in equilibrium are zero in expectation over the joint distribution of agent types. A mechanism is *ex post budget balanced* if the net transfers in equilibrium are zero for all type profiles. These conditions can be relaxed to *weak* budget balance by instead stipulating that net transfers must be non-negative, or formally: $\sum_i p_i(\boldsymbol{\theta}) \geq 0$

Individual Rationality A mechanism is *individually rational* when agents would voluntarily choose to participate in the mechanism. Suppose agents can attain a utility $\tilde{u}_i$ outside of the mechanism. A mechanism implementing social choice function f has *ex ante* individual rationality if agents choose to participate before they know their own type, i.e.:

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta}}\left[u_i(f(\boldsymbol{\theta}), \theta_i)\right] \geq \mathop{\mathbb{E}}_{\theta_i} \tilde{u}_i(\theta_i) \tag{2.6}$$

A mechanism has *ex interim* individual rationality if agents choose to participate after they know their own type θ_i, i.e.:

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta}_{-i}}\left[u_i(f(\boldsymbol{\theta}), \theta_i)\right] \geq \tilde{u}_i(\theta_i) \tag{2.7}$$

This is the most commonly used version. Lastly, a mechanism has *ex post* individual rationality if agents would choose to participate for all agent types $\boldsymbol{\theta}$:

$$u_i(f(\boldsymbol{\theta}), \theta_i) \geq \tilde{u}_i(\theta_i) \ \forall\ \boldsymbol{\theta} \tag{2.8}$$

2.6 Groves Mechanisms

The Groves family of mechanisms is central to much of the recent work in the field of mechanism design. These are efficient and strategyproof direct-revelation mechanisms for agents with quasi-linear preferences described in a series of papers by Vickrey [1961], Clarke [1971] and Groves [1973].

Definition 2.8: *In a **Groves** mechanism, agents are assumed to have quasilinear utility functions $u_i(\kappa, p_i, \theta_i) = v_i(\kappa, \theta_i) - p_i$, where $\kappa \in \mathcal{K}$ is the choice implemented by the mechanism, and p_i is the payment charged to agent i. Given reports $\hat{\boldsymbol{\theta}}$, a Groves outcome function is comprised of a specific* choice *rule $\kappa(\hat{\boldsymbol{\theta}}) : \boldsymbol{\Theta} \to \mathcal{K}$ and a specific* payment *rule $p_i(\hat{\boldsymbol{\theta}}) : \boldsymbol{\Theta} \to \mathbb{R}$. The stipulated choice rule maximizes the total reported value:*

$$\kappa(\hat{\boldsymbol{\theta}}) = \underset{\kappa \in \mathcal{K}}{\operatorname{argmax}} \sum_{i}^{N} v_i(\kappa, \hat{\theta}_i) \tag{2.9}$$

The payment rule charges a quantity independent of the agent's report, $h_i(\hat{\boldsymbol{\theta}}_{\text{-}i})$, less the total value of the economy without the agent present:

$$p_i(\hat{\boldsymbol{\theta}}) = h_i(\hat{\boldsymbol{\theta}}_{\text{-}i}) - \sum_{j=1, j \neq N}^{N} v_j(\kappa(\hat{\boldsymbol{\theta}}), \hat{\theta}_j) \tag{2.10}$$

where $h_i : \boldsymbol{\Theta}_{\text{-}i} \to \mathbb{R}$ is an arbitrary transfer function whose choice determines which member of the Groves family we are implementing.

The h_i function enables the designer to specify a tradeoff between budget balance and individual rationality, as there are strong impossibility results stipulating that we can't have both along with efficiency, as we will see in Section 2.8. This said, in special cases, such as combinatorial auctions, there are Groves mechanisms that are both individually rational and *weak* budget balanced.

Groves mechanisms are particularly important because of the following result due to Green and Laffont and strengthened to a broader class of valuation functions by Holmström [1979]:

Theorem 2.3: *[Green and Laffont, 1977; Holmström, 1979] When agents' possible value functions are smoothly path connected and contain the zero function, then the Groves mechanisms are the only direct revelation mechanisms that are efficient, strategyproof and where there are no payments by losing bidders. Moreover the unique Groves mechanism achieving this property is known as the Clarke Mechanism (described below in Section 2.6.1).*

We state this without proof; see Ausubel and Milgrom [2006] for a succint treatment. The result requires the technical condition that value functions be drawn from a set of *smoothly path connected* functions. A set of functions will have this property if connecting any two of its elements there is a smoothly-parameterized family functions that are themselves set members. More formally, between any two elements $v(\cdot;0)$ and $v(\cdot;1)$, there exists a path $\{v(\cdot;t)|t \in [0,1]\}$ such that v is differentiable in its second argument and $\int_0^1 \sup_x |{}^{\partial v}\!/\!_{\partial t}\, v(x,t)|dt < \infty$. Williams [1999] has extended this result to Bayes-Nash implementation. When coupled with the revelation principle, this means that *any* incentive-compatible and efficient mechanism for *smoothly path connected* valuations and quasi-linear preferences will implement a Groves outcome.

2.6.1 The Clarke Mechanism

The most important of the Groves mechanisms is known as the Pivotal or Clarke mechanism [Clarke, 1971]:

Definition 2.9: *A* ***Clarke mechanism*** *is a Groves mechanism that uses the following transfer function:*

$$h_i(\hat{\boldsymbol{\theta}}_{\text{-}i}) = \sum_{j=1, j\neq N}^{N} v_j(\kappa_{\text{-}i}(\hat{\boldsymbol{\theta}}_{\text{-}i}), \hat{\theta}_j) \tag{2.11}$$

where

$$\kappa_{\text{-}i}(\hat{\boldsymbol{\theta}}_{\text{-}i}) = \operatorname*{argmax}_{\kappa \in \mathcal{K}} \sum_{j=1, j\neq N}^{N} v_k(k, \hat{\theta}_j) \tag{2.12}$$

is the optimal choice for all the agents except agent i.

Informally, the transfer function is the value to all of the agents other than i of the best outcome of the economy without i. This is independent of i's report, and thus meets the conditions for transfer functions in Groves mechanisms.

The Clarke mechanism is attractive, because it achieves both *ex post* Individual Rationality and *weak* budget balance in often-applicable settings (i.e. it just skirts the impossibility results we will see in Section 2.8.), which we can characterize using the following definitions:

Definition 2.10: *A choice set is* ***monotonic*** *if it weakly increases with additional agents.*

This condition ensures that the addition of another agent won't prevent a 'good' solution already available to the other agents from occurring. We also need:

Definition 2.11: *Agent value has* ***non-negative valuation*** *if* $v_i(\kappa_{\text{-}i}, \theta_i) \geq 0$ $\forall\, \boldsymbol{\theta}_{\text{-}i}, \forall\, i$.

This condition requries that any choice that doesn't involve a given agent, will be at least neutral to that agent. With these definitions, the following theorem holds:

Theorem 2.4: *The Clarke mechanism is ex post individually rational, efficient and strategyproof for domains with quasi-linear preferences that have non-negative valuations and where choice sets are monotonic.*

We state this without proof; see Parkes [2001] for a concise treatment. Importantly, an exchange with private values will meet these conditions, and thus be subject to the above theorem. If we add another condition:

Definition 2.12: *A setting has* ***no positive effect*** *when adding an agent i to the optimal outcome is at best neutral to the value of the outcome available to the other agents.*

then we obtain the following theorem:

Theorem 2.5: *The Clarke mechanism has weak budget balance for domains with quasi-linear preferences that have non-negative valuations and no positive effect, and where choice sets are monotonic.*

Again we state the theorem without proof; see Parkes [2001] for a treatment. For our purposes, this theorem is important because it applies to auction settings where all bidders have *free disposal*[1], making Clarke mechanisms very attractive in these settings, at least on paper. We will discuss some of their drawbacks in Section 2.9.1.

2.7 Combinatorial Markets

Throughout most of this thesis, we are concerned with either combinatorial auctions or combinatorial exchange settings; we describe both here.

2.7.1 Combinatorial Auctions

Combinatorial Auctions (CAs) have received a lot of attention in the recent electronic markets literature, as they apply to many important real-world problems, several of which we discussed in Chapter 1. A combinatorial auction facilitates the sale of sets of items to buyers, each of whom expresses value over *bundles* of items. The manner in which agents communicate with the system is expressive enough to enable the specification of complements and substitutes across items. Typically a concise language is used to enable agents to bid on multiple bundles without requiring exponential communication costs. Note, though, that even with a concise language the bidders' problem of providing values for all the bundles of interest may still be computationally difficult without both carefully designed mechanisms and agents. Rather than describe combinatorial auctions formally, we will instead cover their generaliza-

[1] *Free disposal* means that agents have weakly increasing value in the number of items they receive.

tion:

2.7.2 Combinatorial Exchanges

A Combinatorial Exchange (CE) is a two-sided generalization of a combinatorial auction. It is a mechanism that facilitates trade among multiple buyers and sellers, each of whom expresses value over *bundles* of items, and each of whom is able to express both complements and substitutes on the items within bundles. Formally we have:

A CE is a market with multiple units of distinct, indivisible items, $G = \{1, \ldots, k\}$, and multiple agents $N = \{1, \ldots, n\}$, each of whom may be interested in both buying and selling items. Each bidder has a (possibly empty) 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 $x^0_{ij} \in \mathbb{Z}_+$ indicate the number of units of good type $j \in G$ initially held by bidder $i \in N$. The initial allocation x^0_i may be private to agent i. We assume that bidders are truthful in revealing this information, which we motivate by stipulating that participants cannot sell items that they do not actually own (or will pay a suitably high penalty if they do). 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.

Each agent i has a valuation $v_i(\lambda_i) \in \mathbb{R}$ on possible trades $\lambda_i = (\lambda_{i1}, \ldots \lambda_{ik})$, where $\lambda_{ij} \in \mathbb{Z}$ specifies the number of units of item j transferred to agent i. This value can be positive or negative, and represents the *change in value* between the final allocation $x_i^0 + \lambda_i$ and the initial allocation x_i^0. Agents are generally assumed to have quasilinear utility functions, with $u_i(\lambda_i, p_i) = v_i(\lambda_i) - p_i$ given payments $\boldsymbol{p} \in \mathbb{R}^N$ made to the mechanism. The individual payments p_i can be negative, indicating that bidder i may receive a payment for the trade. This implies that bidders are modeled as being risk neutral and assumes that there are no budget constraints. Further, agents are typically assumed to have *free disposal*, meaning 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. We also assume 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.

The agent valuation function is typically expressed in a concise language; however, for expositional purposes in this chapter we adopt the XOR language [Sandholm, 2002a], so that agents just specify the particular trades in which they are interested and the valuation on other trades is induced by free disposal.

2.7.3 Winner Determination

An efficient CE will identify the trade that maximizes the total value across all trades, subject to feasibility constraints (e.g. supply $\geq$ demand). 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 $\boldsymbol{v} = (v_i, v_N)$ and an initial allocation $x^0 = (x_1^0, \ldots, x_n^0)$, the efficient trade λ^*, is defined as follows:

Definition 2.13: *Given CE instance $(\boldsymbol{v}, x^0)$, the efficient trade λ^* solves the following mixed integer program (MIP):*

$$\max_{(\lambda_1,\ldots,\lambda_n)} \sum_i v_i(\lambda_i) \tag{2.13}$$

$$\text{s.t.} \quad \lambda_{ij} + x^0_{ij} \geq 0, \quad \forall i, \forall j \tag{2.14}$$

$$\sum_i \lambda_{ij} = 0, \quad \forall j \tag{2.15}$$

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

Constraints (2.14) ensure that no bidder sells more items than he has in his initial allocation. By free disposal, we can impose strict balance on the supply and demand of goods at the solution in constraints (2.15); 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 if, as in the above formulation, the valuation function v_i explicitly represents a value for each possible trade to bidder i, 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. Represented in this way, this MIP quickly becomes intractable to write down, let alone solve. In Chapter 5, where we construct a novel CE mechanism, we will address this problem by constructing a concise winner determination formulation tied to the *TBBL* bidding language that we propose using (see Section 5.2.1).

While this concise representation allows the specification of large problems, the winner determination problem still requires solving an NP-hard set packing problem. We mitigate this by solving the induced MIP with a state-of-the-art solver (ILog CPLEX 11), which uses an extremely sophisticated branch-and-cut algorithm to solve remarkably large manifestations of the problem.

2.7.4 The Vickrey-Clarke-Groves (VCG) Mechanism

We now consider applications of the *Clarke* mechanism described in Section 2.6.1 to combinatorial markets.

A quick aside on naming: When applied to a CA, the Clarke mechanism is referred to in the literature as the *Generalized Vickrey Auction* (GVA). One might therefore assume that an application to a CE would be called a “Generalized Vickrey Exchange”, but such terminology is not standard in the literature. Following standard practice, we therefore refer to a Clarke mechanism applied to the CE setting by the more general term *Vickrey-Clarke-Groves Mechanism* (VCG) [e.g. Krishna, 2002].

Given reported valuations $\hat{\boldsymbol{v}} = (\hat{v}_i, \ldots, \hat{v}_N)$, the VCG mechanism selects the efficient trade λ^* based on reports, to maximize the total value over all feasible trades, as described in the previous section. Next, let $V^*(\hat{\boldsymbol{v}}) = \sum_i v_i(\lambda_i^*)$ denote the total value (or *surplus*) over all agents in this trade. Given reported valuation functions $\hat{\boldsymbol{v}} = (\hat{v}_1, \ldots, \hat{v}_N)$, the mechanism then collects the following payments from each bidder:

$$p_{\mathrm{vcg},i} = \hat{\boldsymbol{v}}(\lambda_i^*) - (V^*(\hat{\boldsymbol{v}}) - V^*_{\text{-}i}(\hat{\boldsymbol{v}}_{\text{-}i})), \tag{2.16}$$

where $V^*_{\text{-}i}(\hat{\boldsymbol{v}}_{\text{-}i})$ is the total reported value of the efficient trade in the economy without bidder i and where $\boldsymbol{v}_{\text{-}i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$. It is helpful to re-factor the payments as an adjustment $p_i = \hat{v}_i - \Delta_i$ for some *discount*, where $\Delta_i \geq 0$ ensures individual rationality so that no agent has negative utility from participation. Thus, in subsequent chapters we refer to $\Delta_{\text{vcg},i} = V^*(\hat{\boldsymbol{v}}) - V^*_{\text{-}i}(\hat{\boldsymbol{v}}_{\text{-}i})$ as the *VCG discount*.

The VCG mechanism (in a CE environment) is strategyproof and individually rational, but runs at a deficit. Specifically, we can have $\sum_i p_{\text{vcg},i} < 0$, or equivalently that $\sum_i \Delta_{\text{vcg},i} > V^*$, with more discounts to assign than there is available surplus.

Example 2.1: *Consider the following example:*

Agent 1 *Sell A for* -10

Agent 2 *Sell B for* -5

Agent 3 *Swap A for B for* $+5$

Agent 4 *Buy A and B for* $+35$

Agent 5 *Buy A for* $+15$

The value-maximizing trade is for Agents 1 and 2 sell their items to agent 4, producing a surplus of 20. The payments in the VCG mechanism are:

Agent 1 *Pays* $-10 - (20 - 15) = -15$

Agent 2 *Pays* $-5 - (20 - 5) = -20$

Agent 4 *Pays* $35 - (20 - 15) = 30$,

The sum of these payments is -5*, showing that the mechanism can run at a deficit.*

2.8 Myerson-Satterwaite Impossibility Theorem

The work in Chapters 3 and 4 is strongly motivated by the famous Myserson-Satterthwaite Impossibility theorem:

Theorem 2.6: *[Myerson and Satterthwaite, 1983] It is impossible to achieve allocative efficiency, ex ante budget balance and interim individual-rationality, in a Bayes-Nash incentive compatible mechanism, even with quasi-linear utility functions.*

Myerson and Satterthwaite [1983] prove the theorem in a simple setting where a lone seller is offering a single good to a lone buyer. Neither player knows the other's valuation, causing uncertainty about the trade occurring. However, the theorem is general and applies to arbitrarily complex environments that satisfy the stipulated conditions [Williams, 1999]. From a designer's standpoint, it is typically not possible to relax the individual rationality desiderata, as *enforcing* participation is not possible or desirable. Further, relaxing the desire for budget balance is generally also impossible, as most mechanism users are unwilling to subsidize their operation.

However, in restricted settings it may be possible to substitute *weak* budget balance, where the mechanism can generate a surplus but not a deficit. For example, often in a one-sided auction the excess surplus can be assigned to the seller. Thus, for these settings it is posible for the VCG mechanism described in Section 2.7.4 to escape the jaws of the theorem.

However, in general settings, such as the combinatorial exchange setting we introduced in Section 2.7.2, using *weak* budget balance is insufficient to permit efficient and individually rational mechanism construction, as subsidies become necessary. In these settings, we argue that it is actually best to depart from the classic VCG approach and relax the efficiency requirement, not the budget requirement. In this case, the designer considers mechanisms that are budget balanced, *ex post* individually rational, but inefficient. The goal of the social choice function is then to implement an outcome that is *as efficient as possible*. A characterization of the solution(s) to this problem remains an open problem in mechanism design. However, some results are available for domains with restricted agent preferences; see the work of McAfee [1992b], Chu and Shen [2008] and Babaioff and Walsh [2005].

The work presented in Chapters 3 and 4 seeks to construct payment rules that promote truthful reporting, and to choose the maximally truthful rule among several options, respectively. While truthfulness is a desirable property in its own right, one way to interpret this work is as a step in the search for highly efficient CE mechanisms, because the proposed designs implement the welfare-maximizing trade at reports, and thus will achieve efficiency in the limiting case where agents do in fact state the truth. Chapter 5 discusses one carefully constructed mechanism designed within this framework, which we show empirically to be highly efficient, though a proof of *optimal* approximate efficiency remains elusive.

2.9 Resistance to Defection by Coalitions of Agents

For all of its advantages, there are some substantial problems with the VCG mechanism. Oft cited papers by Ausubel and Milgrom [2006] as well as Rothkopf [2007] provide a discussion of these issues in the general case. Expanding on earlier computational work [Sandholm, 2000], Conitzer and Sandholm [2006] describe the particular problems that appear in combinatorial markets. In such settings, the force driving many of the problems is the potential for VCG to produce exceedingly low revenue to the seller. Ausubel and Milgrom provide a simple example of a combinatorial auction of two goods that shows this revenue shortfall:

Example 2.2: *Suppose bidder 1 wants both items for $1, and bidders 2 and 3 want one item each, also for $1. VCG assigns the goods to bidders 2 and 3. But it charges them both $0! Bidder 2 pays nothing (an analysis of bidder 3 is symmetric) because bidders 1 and 3 generate a total value of $1 regardless of whether or not bidder 2 is playing, and VCG charges bidders their marginal impact on the rest of the players.*

This lack of revenue causes several problems, including:

- Sellers may simply want more revenue. This may have contributed to Yahoo!'s and Google's decision to use the GSP mechanism over Vickrey [Edelman et al., 2007].
- Agents may find it advantageous to split their identity and bid using *shill* players [Yokoo, 2006; Yokoo, Sakurai, and Matsubara, 2004].

- Revenue is not monotonic as the number of agents increase.
- Agents may defect from the outcome of the auction, as they may be able to form a coalition of bidders who together with the seller are able to establish an outcome that is of higher utility to these participants.

2.9.1 Core

To avoid these problems, several authors have argued for using *Core* payments instead of VCG payments [Ausubel and Milgrom, 2002; 2006; Day and Cramton, 2008; Day and Milgrom, 2008; Milgrom, 2004; Milgrom and Day, 2008]. To formalize the concept of the *Core* [Gillies, 1953] we first need the following definition:

Definition 2.14: *The* ***coalition value function*** *of a quasi-linear combinatorial auction for* ***coalition*** $S \subset N$ *is:*

$$w(S) = \max_{\lambda \in \mathcal{F}_S} \sum_{i \in S} v_i(\lambda_i) \tag{2.17}$$

If S contains the seller, and 0 otherwise. Here $\mathcal{F}_S$ denotes the set of trades that are feasible to the set S in isolation.

Informally, the coalition value function determines the value that a subset of the agents can generate on their own. Based on this, we can define:

Definition 2.15: *A* ***blocking coalition*** *in a quasi-linear combinatorial auction setting is a set of players S for whom:*

$$w(s) > \sum_{i \in S} v_i(\lambda^*) - p_i \tag{2.18}$$

where λ^* is the trade chosen by the grand coalition, i.e. the full set of players N.

Informally, a blocking coalition is a group of players who have exactly the condition in the third list of problems above: by working together they can produce more utility for themselves outside the auction than within it.

Definition 2.16: ***Core Outcomes***, *are then those which are feasible and which are not blocked by any coalition:*

$$Core(N, w) = \left\{ \langle \lambda, p \rangle \middle| w(N) = \sum_{i \in N} v_i(\lambda_i) - p_i \quad \wedge \right.$$
$$\left. w(S) \leq \sum_{i \in S} v_i(\lambda_i) - p_i \; \forall \; S \subset N \right\} \tag{2.19}$$

Core Payments *p are those payments contained in this core set.*

The core is known to be non-empty for the combinatorial auction setting described here. However, the VCG outcomes are generally outside of the core, and thus core payments are generally not incentive-compatible. This opens the design question of finding the maximally incentive compatible core payments, a topic we will return to in Chapter 3. [2]

[2] A technical condition known as *buyers are substitutes* is necessary and sufficient to have the VCG outcome in the core.; see the work of Parkes and Ungar [2002] for details.

2.9.2 Least-Core and Nucleolus

The core is known to be empty for general CEs. Thus if we wish to enforce core-like constraints, we need a relaxation. A natural relaxation as defined in the work of Yokoo, Conitzer, Sandholm, Ohta, and Iwasaki [2005] is as follows:

Definition 2.17: *[Yokoo et al., 2005] An* ϵ***-Core Outcome*** *is one which is feasible and which is not within* ϵ *of being blocked by any coalition:*

$$\epsilon\text{-}Core(N,w) = \left\{ \langle \lambda, p \rangle \;\middle|\; w(N) = \sum_{i \in N} v_i(\lambda_i) - p_i \quad \wedge \right.$$
$$\left. w(S) \leq \sum_{i \in S} v_i(\lambda_i) - p_i + \epsilon \;\forall\; S \subset N \right\} \quad (2.20)$$

Which enables us to define:

Definition 2.18: *[Yokoo et al., 2005] A* ***Least-Core Outcome*** *is an* ϵ*-Core outcome where for all* $\epsilon' < \epsilon$*, we have that the* ϵ'*-Core set is empty.*

This definition is attractive for two reasons. First, because of the relaxation, the Least-Core is non-empty for all coalitional value functions. Secondly, it is relatively easy to specify through linear programming techniques, as one need only minimize the variable representing ϵ.

However, the set of Least-Core outcomes typically will not be unique, as we place no binding restriction on the coalitional value function of those coalitions whose relaxation constant need not be as large as ϵ. To obtain a unique outcome, we can instead determine the minimal relaxation possible for every coalition in turn, lexicographically from highest to lowest. This is called the *Nucleolus* outcome [Schmeidler, 1969], and is guaranteed to both exist and be unique. For a formal description, see the work of Maschler, Peleg, and Shapley [1979].

2.10 Prices

For the pricing algorithm defined in Section 5.3.4, we will also need properties of *prices* in quasi-linear environments. In this thesis, we mostly talk about the *payments* that agents make for some outcome choice (i.e. a monetary transfer portion of the mechanism outcome rule). But we can also speak about *prices* on outcomes in general, whether or not agents will actually pay these amounts. And mechanisms can quote prices that are quite distinct from the payments that are ultimately made (e.g. in a standard English auction the announced prices are less than the ultimate payments in all rounds save the very last). A *price* is always a single value representing a *potential* monitary transfer. In a combinatorial market setting, we also refer to a *price function* $p(\lambda_i) : \Lambda_i \to \mathbb{R}$, which maps from a trade (e.g. to agent i) to the price for that trade.

2.10.1 Types of Prices

The prices we encounter in everyday use are *anonymous linear*, in that they are additive in each additional unit. Formally we have:

Definition 2.19: *A vector of* ***Anonymous Linear Prices*** ϕ *for a vector of items being traded* λ_i*, result in a payment of* $p(\lambda_i) = \phi \cdot \lambda_i$*. Note that to make clear which price vector we are using to define this pricing function, we often write* *this as $p^{\phi}(\cdot)$.*

Prices can also be agent-specific, giving us:

Definition 2.20: *A matrix of **Non-Anonymous Linear Prices** ϕ_{ij} for a vector of items being traded λ_i result in a payment of $p_i(\lambda_i) = \phi_i \cdot \lambda_i$.*

Prices can also be non-linear, in that they can be super- or sub-additive in the items in a bundle. Formally:

Definition 2.21: ***Anonymous Non-Linear Prices*** *$p : \Lambda_i \to \mathbb{R}$ are bundle specific prices, that assign a payment to each possible bundle of items that could be traded $\lambda_i \in \Lambda_i$.*

And these prices can be made non-anonymous as well:

Definition 2.22: ***Non-Anonymous Non-Linear Prices*** *$p_i : \Lambda_i \to \mathbb{R} \; \forall \; i \in N$ are bundle specific prices, that assign a payment to each possible bundle of items that could be traded $\lambda_i \in \Lambda_i$ separately for each agent.*

2.10.2 General Competitive Equilibrium Prices

Competitive equilibrium prices with respect to a given outcome are those at which each agent weakly prefers his portion of the outcome to all others at the given prices, and the outcome is feasible (e.g. in the sense that supply matches demand) [Mishra and Parkes, 2007]. For quasi-linear bidders, we have:

Definition 2.23: *A (potentially non-anonymous non-linear) price function $p_i(\cdot)$ specifies **competitive equilibrium (EQ)** prices if*

$$v_i(\kappa^*) - p_i(\kappa^*) = \max_{\kappa \in \mathcal{K}} v_i(\kappa) - p_i(\kappa) \tag{2.21}$$

holds for all agents i where κ^* is the outcome being implemented.

With this definition of competitive equilibrium prices we have the following useful theorem, which we will need in Chapter 5.

Theorem 2.7: *[Bikhchandani and Ostroy, 2002] Any trade λ supported by competitive equilibrium prices $p(\cdot)$ is an efficient trade.*

2.10.3 Linear EQ Prices in a Combinatorial Exchange Setting

We can specialize the above definition of competitive equilibrium prices to an exchange setting. See the work of Wurman and Wellman [1999] for a description of how to define non-linear non-anonymous EQ prices in a combinatorial market environment. Here we define anonymous linear EQ prices for a CE setting, as we will need them in Chapter 5.

Linear prices, ϕ, define a price ϕ_j on each good so that the price to bidder i on a trade λ is defined as $p^{\phi}(\lambda_i) = \sum_j \lambda_{ij}\phi_j = \lambda_i \cdot \phi$.

Definition 2.24: *Linear prices ϕ 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:*

$$v_i(\lambda_i) - p^\phi(\lambda_i) \geq v_i(\lambda_i') - p^\phi(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0), \tag{2.22}$$

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

2.10.4 Approximate Linear EQ Prices

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

Definition 2.25: *Linear prices ϕ 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:*

$$v_i(\lambda_i) - p^\phi(\lambda_i) + \delta \geq v_i(\lambda_i') - p^\phi(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0), \tag{2.23}$$

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, we 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 2.8: *Any trade λ supported by δ-approximate EQ prices ϕ is a $2\min(M, \frac{n}{2})$ δ-approximate efficient trade. Here M is the total supply of goods and n is the number of agents participating, as described in Section 2.7.2.*

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

$$\sum_{i \in \lambda \cup \lambda'} [v_i(\lambda_i) - p^\phi(\lambda_i) + \delta] \geq \sum_{i \in \lambda \cup \lambda'} [v_i(\lambda'_i) - p^\phi(\lambda'_i)], \tag{2.24}$$

by δ-EQ prices and because values and prices are zero for bidders that do not participate in a trade. We have $\sum_{i \in \lambda \cup \lambda'} p^\phi(\lambda_i) = \sum_{i \in \lambda \cup \lambda'} p^\phi(\lambda'_i) = 0$ (since $\sum_i p^\phi \lambda''_i = \sum_i \lambda''_i \cdot \phi = \sum_i \sum_j \lambda''_{ij} \phi_j = \sum_j \phi_j \sum_i \lambda''_{ij} = 0$, 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

$$\Delta = \sum_{i \in \lambda \cup \lambda'} \delta_i \leq \min(2A^{\#}(x^0), n)\delta \leq \min(2\min(M, n), n)\delta = 2\min(M, \frac{n}{2})\delta \tag{2.25}$$

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 sufficiently many number of goods to trade or bidders in the market, and thus $A^{\#}(x^0) \leq \min(M, n)$. $\square$

2.11 Summary

In this chapter we have introduced several key formalisms and results from both Game Theory and Mechanism Design. We defined the mathematical construct of a *mechanism* and discussed the *design* thereof. We covered the *solution concepts* under which such designs are evaluated, and the game theoretic equilibria that underpin them. We discussed various properties that we might like both our *social choice functions* and our mechanisms to embody, including *efficiency*, *individual rationality* and *budget balance*. We described the Myerson-Satterwaite impossibility theorem, and its implication that we can't generally achieve these three properties simultaneously. We formally defined the combinatorial market settings that will be investigated in subsequent chapters, and related these to the impossibility result. We described the famous VCG mechanism, and several of its advantages (strategyproofness), as well as several of its drawbacks (often outside the core). We defined *core* outcomes, and why they are desirable, as well as their approximations. And finally, we discussed *competitive equilibrium* prices, and proved a theorem about linear approximations thereof. With this background in hand, we can now turn to an examination of incentives in combinatorial markets.

Truth is truth
To the end of reckoning.

– William Shakespeare
Measure for Measure, Act V, Scene 1

3

Automated Payment Design

The concept of *strategyproofness* was introduced formally in Section 2.4. Informally, a mechanism is strategyproof if it is a dominant strategy equilibrium for every agent to report its private information (or type) truthfully. Strategyproofness simplifies participation and removes the need for counter-speculation about the behavior of other agents. But strategyproofness can be unachievable together with other desirable properties [Myerson and Satterthwaite, 1983]; such properties include budget-balance [Parkes et al., 2001a], coalitional stability or revenue properties [Ausubel and Milgrom, 2006], simple rules [Lahaie et al., 2006], and computational tractability [Nisan and Ronen, 2000; Sandholm, 2002b]. It is thus often necessary to adopt approximately strategyproof mechanisms.

3.1 Motivation

In this chapter we examine the problem of designing mechanisms that are as close to strategyproof as possible in over-constrained settings. Here, as elsewhere in this thesis, we restrict ourselves to direct mechanisms that solicit bids from participants and then choose an outcome according to these reports. Payments are then charged to the participants according to a specific *payment rule*. With this restriction in mechanism structure, the problem under consideration in this chapter reduces to finding the payment rule that provides incentives to the agents that is as close to strategyproof as possible.

The first task in this process is to provide a formal statement about what we mean by *"approximately strategyproof"*. After these definitions, we will turn to a theoretical model for the optimal mechanisms according to these definitions. Next we will discuss how to approximate this ideal rule using Linear Programming techniques, and the challenges therein. Finally several examples of this method will be described. The theory offered in this chapter is applicable to most combinatorial mechanisms though for both expository and computational reasons, the examples will be limited to combinatorial auctions.

3.2 Preliminaries

We will reserve a full discussion of related work until Section 4.9, and instead offer here a description of key previous contributions needed for the content that follows. In this chapter we will be concerned with formulating a mechanism design as a form of constrained optimization. We therefore highlight two important bodies of previous work in this area.

3.2.1 Minimizing the Per Instance Distance to VCG in Settings without Full Strategyproofness

In work published in 2001, Parkes, Kalagnanam, and Eso [2001a; 2001b] proposed to choose rules by finding payments that are as close to the VCG payments described in Section 2.7.4 as possible *within a given market instance.* This goal is one possible definition of minimizing *ex post* regret, in that it targets an agent's utility loss relative to what he could have achieved under a strategyproof mechanism (i.e. VCG). They describe the rules in the context of a budget balanced combinatorial exchange, where VCG payments are unavailable because they run at a deficit. And consequently, when budget balance is enforced, agent *regret* may be quite large. The approach can be adapted for other over-constrained design settings, such as combinatorial auctions with core constraints (where VCG is typically out of the core, and thus unavailable) [Milgrom and Day, 2008]. Specifically, they propose the following program for specifying the discounts used to define the payments:

$$\begin{aligned}
\operatorname*{argmin}_{\boldsymbol{\Delta}} \quad & L(\boldsymbol{\Delta}, \boldsymbol{\Delta}_{vick}) && \text{Min Distance} \quad (3.1)\\
\text{s.t.} \quad & \sum_{i \in I} \Delta_i \leq V^* && \text{Budget Balance} \quad (3.2)\\
& \Delta_i \leq \Delta_{vick,i},\ \forall\, i \in I && \text{Within Vickrey} \quad (3.3)\\
& \Delta_i \geq 0,\ \forall\, i \in I && \text{Individual Rationality} \quad (3.4)
\end{aligned}$$

where $\mathbf{\Delta}$ is the discount vector defining the rule, I is the set of agents trading, and V^* is the total surplus available.

The program defines a different payment rule for each possible distance function $L(\cdot,\cdot)$. Parkes et. al. discuss rules defined by several different distance functions, as summarized in the following table:[1]

Rule Name	Distance Function	Discount Definition	Description
Threshold	L_2, L_∞	$\max(0, \Delta_{vick,i}\text{-}C)$	Allocate surplus to minimize the maximum $\Delta_{\text{vcg},i} - \Delta_i$, subject to $\Delta_i \leq \Delta_{\text{vcg},i}$, $\forall i \in N$
Reverse	$\prod_i \frac{\Delta_{vick,i}}{\Delta_i}$	$\min(\Delta_{vick,i}, C)$	Allocate surplus to maximize the minimum $\Delta_{\text{vcg},i} - \Delta_i$, subject to $\Delta_i \leq \Delta_{\text{vcg},i}$, $\forall i \in N$ (and allocating all of the surplus)
Small	$\sum_i \frac{\Delta_{vick,i}-\Delta_i}{\Delta_{vick,i}}$	$\Delta_{vick,i}$ if $\Delta_{vick,i} \leq C$	Allocate surplus from smallest $\Delta_{\text{vcg},i}$ to largest, never exceeding $\Delta_{\text{vcg},i}$
Large	$\sum_i \Delta_{vick,i} (\Delta_{vick,i}\text{-}\Delta_i)$	$\Delta_{vick,i}$ if $\Delta_{vick,i} \geq C$	Allocate surplus from largest $\Delta_{\text{vcg},i}$ to smallest, never exceeding $\Delta_{\text{vcg},i}$
Fractional	$\prod_i \frac{(\Delta_{vick,i})^2}{\Delta_i}$	$\mu\Delta_{vick,i}$	Allocate surplus in proportion to VCG discounts
Equal	-	$\frac{V^*}{\|I\|}$	Split surplus equally among the trading agents
No Discount	-	0	Each agent pays its reported value

Here the constant C is chosen to ensure that constraints 3.2-3.4 are met. Parkes et al. identify Threshold as the most desirable of these rules, noting that it minimizes the

[1] **Equal** and **No Discount** are two additional simple rules used for comparison purposes. Note that **Equal** may provide more than the VCG discount to an agent

ex post maximal incentive to manipulate when other agents play truthfully. In this work, we use these rules as benchmarks when evaluating novel rules.

We likewise consider the *ex post* maximal incentive to deviate, but will argue that other design objectives are more important. In particular, the *ex post* maximal incentive is both a full information condition because agents have complete knowledge of their opponents reports and a worst case condition because these reports are taken to be those that provide the maximal incentive for strategic behavior. In Section 3.3 we discuss alternative assumptions. Then in Chapter 4 we leverage this broader perspective to advocate for a metric that captures approximate strategyproofness based on distributional properties of payments across instances rather than per instance, and show that this broader approach aids the identification of mechanisms with better equilibrium behavior.

3.2.2 Automated Mechanism Design

Automated mechanism design (AMD) is a methodology put forward by Conitzer and Sandholm that formulates a particular class of mechanism design problems as a type of constrained optimization [2006; 2002; 2003a; 2003b; 2004b; 2007]. Their construction searches for a complete outcome function (e.g. a trade and payments) simultaneously. The objective of the optimization is some goal of the designer, e.g. social welfare; constraints are imposed on the solution, typically full strategyproofness and individual rationality. The formulation is exponential in the number of agents, and as such simplifications are needed to use the method even for simple cases. No concept of approximate strategyproofness is considered; mechanisms are formulated to support dominant strategy or Bayes-Nash equilibria and the search is with respect to some other desirable property, e.g. social welfare or revenue. A more complete discussion of previous work on AMD is provided in Section 4.9.3.2.

3.3 Defining Approximate Incentive Compatibility

3.3.1 Bayes-Nash Incentive Compatibility as a Direct Target

Bayes-Nash Equilibrium (BNE) is the best formal model available today for the behavior of self-interested agents in incomplete games where there is no *ex post* or dominant strategy equilibrium available, even though its assumptions of rationality are strict (e.g. requiring common knowledge of types) and it's not clear that people actually play such equilibrium strategies. Not withstanding this concern, BNE provides as the basis for a well defined concept of what it would mean to be approximately strategyproof. Specifically, one might want to consider $D(\boldsymbol{v}||\hat{\boldsymbol{v}}_{BN})$, or some distance D between agents' true values and their reports in the Bayes-Nash equilibrium of the mechanism. But there are several reasons why this is a bit problematic.

First, as soon as we are talking about an equilibrium condition, we are no longer concerned with a single instance, and thus the description isn't complete without a decision about the distribution of $\boldsymbol{v}$. Even accounting for this issue, the criterion doesn't establish a direct link between the detailed definition of the mechanism and its strategic properties (without first going through an equilibrium calculation), making it very difficult to use in driving design. Moreover, because we must calculate the equilibrium, the condition is computationally intractable in even marginally complex settings; we need a method that avoids this computation in order to tackle settings such as combinatorial exchanges. Finally, even within its scope, this distance may not be the right target for a design. While a surplus maximizing outcome function in a strategyproof mechanism will immediately be *efficient*, the ordering provided by the distance metric on $\hat{\boldsymbol{v}}_{BN}$ may not be the ordering with respect to underlying allocative efficiency in equilibrium.

Due to the complexity of calculating BNE, we face a choice. We can stick within the direct method of an equilibrium analysis and be forced to make simplifying assumptions about the complexity of the mechanism (see Empirical Mechanism Design [Vorobeychik and Wellman, 2008] as described in Section 4.9). Or we can turn to an indirect method for defining approximate incentive compatibility where we will identify quantities that are designed to be predictive of the deviation from truth, defined using assumptions about information and risk that imply distributional notions that are appropriate to our performance goals. In this chapter, we characterize optimal mechanisms under such an indirect goal. In Chapter 4, we will turn to a more computationally simple method that doesn't require complex optimization, forming a bridge between both approaches in Section 4.7.

Next we will define the specifics of several indirect definitions of approximate incentive compatibility that operate by bounding the potential gain from manipulation.

3.3.2 Bounding the Gain from Manipulation

There are a number of different ways to define the possible gain from manipulation under a given mechanism. The definitions offered here will be used both in this chapter and in Chapter 4.

We consider a mechanism mediating among a set of bidders $N = 1, ...n$ each with a general independent valuation function $v_i(\lambda_i) : \Lambda_i \to \mathbb{R}$ drawn from a space V_i, together $\boldsymbol{v}$. Each of these is defined over some finite and feasible outcome $\lambda_i \in \Lambda_i$, where λ_i is the part of outcome λ that pertains to agent i.[2]

To simplify the exposition, instead of characterizing v_i as a function, we will instead consider it a vector with one dimension for each outcome in Λ_i (i.e. $v_i \in V \equiv \mathbb{R}^{|\Lambda_i|}$. This construction simplifies the analysis as it enables outcome and payment rules to be defined as vector valued functions instead of functionals, as we will see below and again in Section 3.4. To be clear, the valuation vector $v_i \in V_i$ is general, and defines $v_i(\lambda_i) \in \mathbb{R}$ for all trades $\lambda_i \in \Lambda_i$, and independently of the trades of other agents.

We denote the function that chooses the outcome, not including the payments, as $\lambda = \mathcal{W}(\boldsymbol{v})$, and note that here we do not require this choice to be surplus maximizing (though this is a likely design decision). We take this function to be both fixed and given. A payment function $p : \mathbb{R}^{|\Lambda|} \to \mathbb{R}^n$ from space $\mathcal{P}$ is a vector valued function from the supplied bids to the payments charged to each agent. We denote the i^{th} component of this function by p_i. Here we are taking advantage of the vector

[2]The definitions in this section can be generalized to infinite outcome spaces, but we focus on finite spaces for purposes of exposition.

representation of the bids to avoid having to describe the payment as a vector-valued functional. It is the choice of this function that will determine the strategic properties of the mechanism being examined.

For convenience, let's define π_i to be agent i's profit under the payment rule p:

$$\pi_i(\boldsymbol{v}) = v_i(\mathcal{W}(\boldsymbol{v})) - p_i(\boldsymbol{v}) \tag{3.5}$$

Further, when agent i has true value v_i and instead reports v_i', let's define his profit under payment rule p as:

$$\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i}) = v_i(\mathcal{W}(\langle v_i'; \boldsymbol{v}_{\text{-}i}\rangle)) - p_i(\langle v_i'; \boldsymbol{v}_{\text{-}i}\rangle) \tag{3.6}$$

where the semicolon syntax indicates a vector composition in the joint bid space using the specified bid for agent i.

3.3.2.1 Expected *Ex Ante* Gain from Manipulation

We can define the *ex-ante* expected incentive for agent i to misreport, $\epsilon_{EA,i}$ as:

$$\epsilon_{EA,i} = \mathop{\mathbb{E}}_{v_i}\left[\max_{v_i'}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_i(\boldsymbol{v})\right]\right]\right] \tag{3.7}$$

The definition is *ex ante* in the sense that the agent only has probabilistic information about the other agents' reports when choosing his misreport v_i', which is consistent with agents being poorly informed about other agents' values in the market being evaluated. Further it is an expected case, in the sense that an expectation is taken over agent i's own type.

This criterion is well-defined with respect to a joint prior that is defined within the BNE of the mechanism. However, we typically think of the criteria as being evaluated at the truthful joint distribution of value. In this case it represents a *unilateral* incentive to deviate away from that truth. Where the mechanism is "reasonably" strategyproof, the two distributions will be similar, and the criterion will measure a similar value.

Such an expectation could be critiqued in that agents generally know their own type. However, if we consider the use of the bound by a designer that needs to minimize the incentives to deviate across a population of potential market instances, then such an expectation with respect to agent types is well motivated. A designer willing to tolerate possibly extremely bad worst-case behavior, may well be able to achieve superior expected case incentive properties by minimizing this quantity. The better the designer's prior information on agent value is, the better it will be able to do in this expected sense.

3.3.2.2 Worst Case *Ex Ante* Gain from Manipulation

Next we consider the worst case deviation across agent i's possible values:

$$\epsilon_{WA,i} = \max_{v_i} \left[\max_{v_i'} \left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}} \left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i}) \right] - \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}} \left[\pi_i(\boldsymbol{v}) \right] \right] \right] \tag{3.8}$$

Note that this is equivalent to equation (3.7) where agent i's value distribution is replaced with a Dirac delta function, peaked at the value that creates the largest incentive to misreport. A designer minimizing this quantity may forego low misreports in expectation, in order to achieve good incentives in the worst case. Further, in the special case where the $\boldsymbol{v}_{\text{-}i}$ distribution is obtained in equilibrium instead of truth, then as this bound goes to zero, we will have full Bayes-Nash incentive compatibility.

3.3.2.3 The Middle Ground: A Quantile

We can create a measure that is a middle ground between equations (3.7) and (3.8), namely the possible gain at a specific quantile.[3] More formally, let

$$g_i(v_i) = \max_{v_i'} \left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}} \left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}} \left[\pi_i(\boldsymbol{v})\right] \right] \tag{3.9}$$

be the available gain when the true value for agent i is v_i. Further, if f_{v_i} is the probability density function of agent i's value, then we can do a change of variables to get a density for the available gain f_{g_i}:

$$f_{g_i}(\epsilon) = \left| \frac{1}{g_i'(g^{\text{-}1}(\epsilon))} \right| \cdot f_{v_i}(g^{\text{-}1}(\epsilon)) \tag{3.10}$$

With the corresponding cumulative distribution function:

$$F_{g_i}(\epsilon) = \int_{-\infty}^{\epsilon} f_{g_i}(\epsilon) d\epsilon \tag{3.11}$$

We then consider the value of a particular quantile of this distribution according to the quantile function:

$$F_{g_i}^{\text{-}1}(q) = \inf \left\{ \epsilon \in \mathbb{R} : q \leq F_{g_i}(\epsilon) \right\} \tag{3.12}$$

where q is a particular quantile, i.e. .5 for the median, or 1 to produce the same result as in equation (3.8). Generally we will be interested in values such as as $q = .75$. Thus, a given payment rule p will induce an implied $F_{g_i}^{\text{-}1}(q)$.

This approach is extremely appealing in that it allows us to focus on the median case, worst case, or any other quantile in between. Because the median is always within one standard deviation of the mean, it also provides a reasonable bound on the expected case as well.

[3]Such quantiles are called quartiles when discretized into 4 bins, deciles for 10 bins, or percentiles for 100 bins.

3.3.2.4 Expected *Ex Post* Gain from Manipulation

We can also consider the case where agents are fully informed about the values of other agents, and can thus optimally manipulate them based upon this information. We might do this both because it is thought to be a better model of the environment, and for reasons of computational tractability. In this case, equation (3.7) becomes:

$$\epsilon_{EP,i} = \mathop{\mathbb{E}}_{v_i}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\max_{v_i'}\left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i})\right] - \pi_i(\boldsymbol{v})\right]\right] \tag{3.13}$$

This formula reflects the agent's expected regret across possible joint types: i.e., the *ex post* gain he might have realized had he misreported optimally when we take an average over the possible joint values.

3.3.2.5 Worst Case *Ex Post* Gain from Manipulation

We can define *ex-post* maximal incentive for agent i to misreport when taken as a worst case over the full joint value space:

$$\epsilon_{R,i} = \max_{\boldsymbol{v}_{\text{-}i}} \max_{v_i} \max_{v_i'}\left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i}) - \pi_i(\boldsymbol{v})\right] \tag{3.14}$$

This is the standard *ex post* maximal regret that is often considered in the literature, and will be zero in a dominant strategy equilibrium, should the mechanism admit one.

The first thing to notice about this condition is that it represents a worst case over the *other* agent valuations. This makes sense in settings that admit a full dominant-strategy incentive-compatible solution that can achieve strategyproofness in even the very worst case. But its' not well motivated for settings where we know there will be instances that permit gains from deviation. In these cases, we may not wish to be focusing on this very worst case, especially as agents may not be perfectly informed, and as doing so may leave much room for improvement in more typical cases.

3.3.2.6 The Middle Ground: A Quantile

As we shall see in Section 4.8, there is strong reason to believe that focusing on *either* expected or worst case manipulations may not be ideal, and that a criterion that seeks a middle ground may yield mechanisms that are both reasonably robust and nearly optimal in the common case. We can also define a quantile criterion in a way that is similar to the *ex ante* case, although it turns out to be a bit more complicated.

We start by defining a new function that is the equivalent to equation (3.9) as:

$$g_i(\boldsymbol{v}) = \max_{v_i'} \left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i})\right] - \pi_i(\boldsymbol{v}) \tag{3.15}$$

This is the available gain to agent i; when the joint value profile is $\boldsymbol{v}$ we need the full joint profile, as agents have access to it under an *ex post* information set. However, requiring g_i to be vector-valued makes the following analysis far more difficult:

If $f_{\boldsymbol{v}}$ is the true probability density function of the joint value profile, then we need to perform a multivariate change of variables over a *non*-bijective function g_i to get a pdf of the available *ex post* gain f_{g_i}:

$$f_{g_i}(\epsilon) = \int_{\boldsymbol{v} \in g_i^{-1}(\epsilon)} \frac{f_{\boldsymbol{v}}(\boldsymbol{v})}{\sqrt{\sum_{j \in N} (\frac{\partial g_i}{\partial v_j}(\boldsymbol{v}))^2}} d\boldsymbol{v} \tag{3.16}$$

This integral calculates the probability mass in the image of a given quantity of gain ϵ according to the inverse of g_i function.[4] The numerator of the large fraction is a

[4]Equation (3.16) is included for the interested reader, but note that we do not require evaluating this integral for our use of this criterion in practice.

normalizing constant, to ensure that our derived PDF integrates to 1.

We can now define the corresponding cumulative distribution function:

$$F_{g_i}(\epsilon) = \int_{-\infty}^{\epsilon} f_{g_i}(\epsilon) d\epsilon \tag{3.17}$$

We then consider the value of a particular quantile of this distribution according to the quantile function:

$$F_{g_i}^{-1}(q) = \inf \{\epsilon \in \mathbb{R} : q \leq F_{g_i}(\epsilon)\} \tag{3.18}$$

While the above derivation is not amenable to easy analysis, calculating quantiles is straightforward empirically. Thus despite this complexity, we find the *ex post* quantile critera extremely appealing as it's both easy to compute and, as we shall see in Section 3.7.2, the $F_{g_i}^{-1}(q)$ function can be a very useful way to understand the incentive effects of a payment rule.

3.4 Adopting These Bounds as Objectives for Design

Let's assume that the mechanism designer has a prior joint distribution $f_{\boldsymbol{v}}$ on the valuations, so expectations over possible bids are well defined. Let's also assume for the moment that these critera have merit when used for design. The ideal expected case *ex ante* payment function (in the sense of Section 3.3.2.1, and adopted here as an example) is then the solution to the following stochastic non-linear functional program (i.e. a program where the optimization finds a function, not a point):

$$\underset{p}{\operatorname{argmin}} \quad \max_i \mathop{\mathbb{E}}_{v_i} \max_{v_i'} \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}} \left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i}) - \pi_i(\boldsymbol{v})\right] \qquad \text{Min } \mathbb{E} \text{ Gain} \tag{3.19}$$

$$\text{s.t.} \quad p_i(\boldsymbol{v}) \leq v_i(\mathcal{W}(\boldsymbol{v})) \quad \forall\, i, \boldsymbol{v} \qquad \text{IR} \tag{3.20}$$

$$\sum_i p_i(\boldsymbol{v}) \geq 0 \quad \forall\, \boldsymbol{v} \qquad \text{Weak BB} \tag{3.21}$$

The objective (3.19) minimizes the maximum expected profit from deviation from v_i to v_i' subject to individual rationality (3.20), and weak budget balance (3.21). In this expression, $v_i, v_i', \boldsymbol{v}_{\text{-}i}, \boldsymbol{v}$ all specify full bids over multiple possible trades, and are value vectors in a space with a dimension for each trade. This enables us to characterize the payment as a vector-valued function over these vectors, instead of having to treat it as a vector-valued functional. Similar programs can be specified for the other definitions of gain discussed above, and various additional or alternate constraints might be imposed depending on the mechanism setting (e.g. core constraints, etc.).

One might be tempted to try to solve for the payment function by using the calculus of variations [Dacorogna, 2008; Jost and Li-Jost, 1998] to minimize the functional prescribed in the objective (and note that each of the gain criteria described earlier corresponds to one such functional). However, this is analytically intractable for several reasons. Most importantly, the *max* that occur outside of the expectations are not consistent with classical formulations of the calculus of variations. Secondly, the complicated winner determination process embedded within the functional is not easily specified analytically. Moreover, even if such an analysis were possible, its result would be a system of partial differential equations that themselves would be very difficult to solve. We are thus left to approximate this program via numerical methods.

3.5 Instantiating this Agenda to Combinatorial Auctions and Exchanges

With careful statistical sampling, discretization and formulation, we can create a good LP approximation to the above intractable functional program. This section describes one such approach. To this end, we now specialize to a combinatorial market setting, as described in Section 2.7.2. We denote those trades which are *feasible* given the initial endowment $\boldsymbol{x}^0$ as $\mathcal{F}_{\boldsymbol{x}^0} \subseteq \mathbb{Z}^{n\times m}$. Similarly, we use $\mathcal{F}_{\boldsymbol{x}^0,i}$ for the subset that are feasible to agent i. We hereafter omit the implied $\boldsymbol{x}^0$ subscript. The mechanism will choose the *efficient* trade $\lambda^* = \mathcal{W}(\boldsymbol{v}, \boldsymbol{x}^0)$ where $\boldsymbol{v} = (v_1, ...v_n)$ and $\boldsymbol{x}^0$ is the endowment used to define $\mathcal{F}$, here after omitted.

According to the joint prior on valuations, certain trades in $\mathcal{F}$ are more likely to occur then others. We can calculate the marginal probability of the occurrence of a given trade λ as:

$$f_\lambda(\lambda) = \int_{\boldsymbol{v}} \boldsymbol{I}_{[\mathcal{W}(\boldsymbol{v})=\lambda]} f_{\boldsymbol{v}}(\boldsymbol{v}) d\boldsymbol{v} \tag{3.22}$$

Likewise, a similar equation defines the marginal probability of the occurrence of a given trade to a given agent $f_{\lambda_i}(\lambda_i)$.

We then let $\mathcal{G} \subseteq \mathcal{F}$ be the set of trades most likely to occur according to f_λ; the size of this set we leave as a parameter to our approximation.[5] We then augment this set with the optimal trade at the actual bids: $\mathcal{G}^* = \mathcal{G} \cup \{\lambda^*\}$. Let $\mathcal{G}_i^*$ be those trades in $\mathcal{G}^*$ where agent i trades.

[5] It is acceptable to let $\mathcal{G} = \emptyset$, which will significantly simplify what follows, at the expense of fidelity. Regardless, we expect $|\mathcal{G}|$ to be small.

3.5.1 Discretizing the Bids

Next we need to define discrete versions of the agent's value function which we can use to drive our LP formulation. To do this we specify a set of bins that will be associated with each possible agent trade under consideration, $\Xi \subset \mathbb{Z}_+$ (e.g. $\{1 = \text{Low}, 2 = \text{Medium}, 3 = \text{High}\}$).[6]

There are many different ways one might establish the bin boundary locations for these bins, and we can choose to adopt any such method. Here we describe one good possible choice, which is also the one used in the experiments presented later in the chapter: systematic *inverse transform sampling* [Devroye, 1986; Steinbrecher and Shaw, 2008]. This will put equal probability mass on each bin for a given agent trade. More formally, the boundary below (1 indexed) bin ξ for trade λ_i and agent i will be given by:[7]

$$\underline{b}_i(\lambda_i, \xi) = F^{-1}_{\lambda_i}\left(\frac{\xi - 1}{|\Xi|}\right) \tag{3.23}$$

Where $F^{-1}_{\lambda_i}$ is the quantile function of the agent specific trade distribution. Similarly, the boundary above is given by:

$$\overline{b}_i(\lambda_i, \xi) = F^{-1}_{\lambda_i}\left(\frac{\xi}{|\Xi|}\right) \tag{3.24}$$

If the reported bids are $\hat{\boldsymbol{v}}$, then we define the reported bins as $\hat{\mathbf{v}} \in \mathbf{V} : \hat{\boldsymbol{v}} \in [\underline{b}(\hat{\mathbf{v}}), \overline{b}(\hat{\mathbf{v}})]$. For these bins we choose the actual values, $\not{b}(\lambda_i, \xi) = \hat{\boldsymbol{v}}$, as a representative point. Note that this means that a distinct version of the LP needs to be solved for any given market instance, as we customize the representative points for those bins that

[6]The size of this set is an approximation parameter.

[7]For clarity of exposition, we omit the agent subscript to the boundary constants $\underline{b}_i$, $\not{b}_i$ and $\overline{b}_i$ where it is clear from context

correspond to the reports. This stands in contrast to AMD which doesn't customize its optimization problem to a specific instance. For all the other bins, we also choose a representative point given by the bin median:[8]

$$\not{b}_i(\lambda_i, \xi) = F^{-1}_{\lambda_i}\left(\frac{\xi - 1/2}{|\Xi|}\right) \tag{3.25}$$

In the bins containing the reported values, we use these values as the representative points directly, in order to ensure such points satisfy the Individual Rationality conditions. However, if the bin size is large, then this dependence on the reported values may introduces a small additional manipulative opportunity. One way to mitigate this potential problem is to slightly relax the IR constraints and use the median point for all of the bins, including those containing reports. If no IR relaxation is permissible and large bin sizes are still required, then a compromise can be obtained by using $\min(\hat{v}_i, \not{b}_i(\lambda_i, \xi))$, which will always satisfy IR, but uses the non-manipulable median point where possible.

With the bin boundaries defined, we can define the space of discretized bids agent i can make among the subset of trades that we are considering. In particular, by assuming a fixed ordering of these trades we can represent a discretized version of bid $v_i \in V \equiv \mathbb{R}^{|\mathcal{F}_i|}$ as $\mathtt{v}_i \in \mathtt{V}_i \equiv \Xi^{|\mathcal{G}^*_i|}$. Note the dimensionality reduction here: we are collapsing the trades in $\mathcal{F}_i - \mathcal{G}^*_i$ into an omitted equivalence class. By construction, we then have:

$$\underline{b}(\lambda_i, \mathtt{v}_i[\lambda_i]) \leq v_i[\lambda_i] \leq \overline{b}(\lambda_i, \mathtt{v}_i[\lambda_i]) \;\forall\; \lambda_i \in \mathcal{G}^*_i \tag{3.26}$$

In subsequent sections we drop the bracketed indexes when referencing the bin boundaries so that e.g. $\underline{b}(\lambda_i, \mathtt{v}_i[\lambda_i])$ becomes $\underline{b}(\lambda_i, \mathtt{v}_i)$.

[8] Other choices for the representative points are possible.

It is worth pointing out that this method of reducing the trades under consideration to only those that are most likely to occur is distinct from the methods used in constrained AMD. Here we gain tractability by limiting the fidelity of the environment being designed for (by limiting the trades under consideration), and through discretization of the payment function. Implementations of constrained AMD by contrast generally gain tractability by limiting the mechanisms being designed to a particular parametrized subclass known to have desirable properties (e.g. known to be completely strategyproof), and then attempting to optimize only over this subclass [Likhodedov and Sandholm, 2005].

We note that the determination of the $\mathcal{G}^*$ set is a second dependency of the LP on the current market instance, so we must run the optimization on every instance not only because the representative points can be instance-specific, but also because the decision as to which trades are included can be instance-specific as well.

3.5.2 Defining Distributions Over the Bins

Now that we have discretized the value space, we will need to represent our prior in terms of this discretization. First we specify the probability of a sample bid from agent i occurring:

$$f_{\mathtt{v}_i}(\mathtt{v}_i) = \int_{\boldsymbol{v}_{\text{-}i}} \int_{v_i \in \Omega(\mathtt{v}_i)} \boldsymbol{I}_{[\mathcal{W}_i(\boldsymbol{v}) \subseteq \mathcal{G}^*_i]} f_{\boldsymbol{v}}(\boldsymbol{v}) dv_i d\boldsymbol{v}_{\text{-}i} \tag{3.27}$$

Where $\Omega(\mathtt{v}_i)$ indicates that portion of agent i's bid space that is covered by $\mathtt{v}_i$ according to equation (3.26).

Next we specify the probability of being in a particular discretization of other agents' bids, given that agent i has made a given discretized bid:

$$f_{\mathtt{v}_{\text{-}i}}(\mathtt{v}_{\text{-}i}|\mathtt{v}_i) = \frac{1}{f_{\mathtt{v}_i}(\mathtt{v}_i)} \int_{\boldsymbol{v}\in\Omega(\mathtt{v})} \boldsymbol{I}_{[\mathcal{W}_i(\boldsymbol{v})\subseteq\mathcal{G}_i^*]} f_{\boldsymbol{v}}(\boldsymbol{v}) d\boldsymbol{v} \tag{3.28}$$

where $\Omega(\mathtt{v})$ indicates that portion of the joint bid space that is covered by bins indicated by $\mathtt{v}$.

3.5.2.1 Inverse Transform Sampling as a Copula

In the special case where the bin boundaries have been chosen by the particular method we proposed in Section 3.5.1, the bins for a given trade will have equal marginal probability mass by construction. That is, this choice of discretization converts the joint value distribution into a copula [Nelsen, 1999]. Consequently, because equation 3.27 is the marginal of this copula it will reduce to a uniform distribution giving us:

$$f_{\mathtt{v}_i}(\mathtt{v}_i) = \frac{1}{|\Xi|^{|\mathcal{G}_i^*|}} \tag{3.29}$$

The joint distribution (i.e. the copula itself) is not similarly constrained, and thus neither is the conditional distribution in equation 3.28.

To build up these probability distributions empirically, as we do for the experiments in sections 3.6 and 3.7, requires care. Typical instance generators, such as those in CATS [Leyton-Brown, Pearson, and Shoham, 2000], do not make a distinction between randomly drawing a specification of the environment from which valuation samples are drawn, and the drawing of the samples themselves. For the method proposed here,we need the value samples to be drawn from a distribution with consistent agent identities, endowments, buyer interest in items (or 'demandments'), and any other data that is specific to the environment, e.g. common values, geographic data etc. To overcome this, the generators we use to drive our experiments first draw a random 'world' that is held constant. Then an arbitrary number of valuation instances that are applicable to this world can be drawn, and it is these that are used to create our empirical distributions.

3.5.3 The Linear Program Formulation

With the definitions in the preceding sections, we are now in a position to construct an LP approximation to the ideal rule by assuming that only trades in $\mathcal{G}^*$ will occur. We implement the payment vector $\overrightarrow{p}_{\hat{\mathtt{v}}}$ derived from the following program (described in detail below):

$$\text{argmin} \quad C_\gamma \gamma + C_\delta \delta + C_\beta \beta \qquad \text{Core, IC, Balance} \tag{3.30}$$

$$\text{s.t.} \quad \delta = \max_{i \in N} \delta_i \qquad \text{Max over agents} \tag{3.31}$$

$$\delta_i = \sum_{\mathtt{v}_i \in \mathtt{V}_i} \delta_{i,\mathtt{v}_i} f_{\mathtt{v}_i}(\mathtt{v}_i) \qquad \text{Expected value} \tag{3.32}$$

$$\delta_{i,\mathtt{v}_i} = \max_{\mathtt{v}'_i \in \mathtt{V}_i \setminus \mathtt{v}_i} \delta_{i,\mathtt{v}_i,\mathtt{v}'_i} \qquad \text{Max over } i\text{'s report} \tag{3.33}$$

$$\delta_{i,\mathtt{v}_i,\mathtt{v}'_i} = \sum_{\mathbf{v}_{\text{-}i} \in \mathbf{V}_{\text{-}i}} f_{\mathbf{v}_{\text{-}i}}(\mathbf{v}_{\text{-}i} | \mathtt{v}_i) \begin{bmatrix} \pi(\mathtt{v}_i, \mathtt{v}'_i, \mathbf{v}_{\text{-}i}) \\ -\pi(\mathtt{v}_i, \mathtt{v}_i, \mathbf{v}_{\text{-}i}) \end{bmatrix} \qquad \text{Expected other bids} \tag{3.34}$$

$$\pi(\mathtt{v}_i, x, \mathbf{v}_{\text{-}i}) \rightarrow \begin{cases} \not{b}(\lambda^x, \mathtt{v}_i) - p_{x;\mathbf{v}_{\text{-}i},i} & \text{if } \lambda^x \subseteq \mathcal{G}^*_i \\ 0 & \text{otherwise} \\ \text{where } \lambda^x = \mathcal{W}_{\not{b}}(x; \mathbf{v}_{\text{-}i}) & \end{cases} \qquad \text{Profit definition} \tag{3.35}$$

$$\gamma + \sum_{i \ni \mathcal{W}_\flat(\mathbf{v}) \wedge i \notin C} p_{\mathbf{v},i} \geq \sum_{i \in C} \begin{bmatrix} \flat(\mathcal{W}_\flat(\mathbf{v}_C), \mathrm{v}_i) \\ - \flat(\mathcal{W}_\flat(\mathbf{v}), \mathrm{v}_i) \end{bmatrix} \forall \mathbf{v}, C \in \wp(N) \qquad \text{Core} \quad (3.36)$$

$$\zeta + \beta \geq p_{\hat{\mathbf{v}},i} \geq \zeta - \beta \quad \forall\, i \qquad \text{Reported Balance} \quad (3.37)$$

$$p_{\mathbf{v},i} \leq \flat(\mathcal{W}_\flat(\mathbf{v}), \mathrm{v}_i) \quad \forall\, \mathbf{v}, i \qquad \text{Individual Rationality} \quad (3.38)$$

$$\sum_i p_{\mathbf{v},i} = 0 \quad \forall\, \mathbf{v} \qquad \text{Budget Balance} \quad (3.39)$$

$$\beta, \delta, \gamma \in \mathbb{R}_+, \delta_*, p_*, \zeta \in \mathbb{R} \qquad \text{Variable Domains} \quad (3.40)$$

$$C_\gamma \gg C_\delta \gg C_\beta \qquad \text{Constants} \quad (3.41)$$

Where $\mathcal{W}_\flat(\mathbf{v})$ is the optimal trade at the representative values $\flat$ for the bins in $\mathbf{v}$, $i \ni \mathcal{W}(\boldsymbol{v})$ denotes that agent i trades at the efficient solution to bids $\boldsymbol{v}$, $\mathbf{v}_C$ is the value profile with only coalition C present, and $\wp()$ denotes the power set.

We note that the formulation does not require the input values $\hat{\boldsymbol{v}}$ to be truthful. However, the input values are used to focus the fidelity of the solution through the selection of $\mathcal{G}^*$ and through the Balance and Individual Rationality constraints.

3.5.3.1 The Objective

The program tries to achieve three potentially competing goals in equation (3.30). Firstly, it minimizes a relaxation to the core constraints γ. For environments where the core constraints can be met exactly, this relaxation can be omitted. For the others, we target the *Least-Core* as a simple approximation for the Nucleolus prices described in Section 2.9.2 [Yokoo et al., 2005]. Secondly, the program minimizes the discretized incentive to deviate from truthful bidding according to the available information, δ. Lastly, it minimizes the distance between the charged payments, β, as a tie-breaking measure. The condition in equation (3.41) ensures that we pick payments for proximity to core first, then for their incentive to deviate, and lastly to break ties. The relative size of C_γ and C_δ could be reversed in environments where coalitions were known to be unable to form.

3.5.3.2 Incentive Compatibility

First, we minimize the maximal incentive to deviate across the various agents in equation (3.31). Next we take advantage of our prior on agent i's value, $f_{\mathtt{v}_i}$, to specify the expected profit from deviation in equation (3.32). In equation (3.33) we consider the alternative reports that agent i might make, choosing the one that maximizes his profit from deviation. Then in equation (3.34) we define this profit from deviation, by taking the average over all the other agents' bids according to our prior on the conditional distribution $f_{\mathbf{v}_{\text{-}i}}$. In the course of specifying the profitable deviation, we need to calculate the available profit for a given agent reporting x when his true value is in $\mathtt{v}_i$ and everyone else has bid $\mathbf{v}_{\text{-}i}$. The formula for this discretized profit is given in equation (3.35), and takes advantage of the representative points for each bin.

We note that when constructing the LP, certain combinations of counter-factual represenrive values will cause a given agent to fail to win his bundle in equation (3.35). In these cases the corresponding payment is known to be zero ahead of time (via IR), and can thus be omitted from the formulation. In fact, when the formulation is for worst-case analysis instead of an expected-case analysis (e.g. equation (3.8) instead of equation (3.7)), careful examination of the relative sizes of the representative points can show constraints corresponding to winning counter-factuals in equation (3.35) as dominated. This is similar to the omission of dominated values from formulations of the AMD problem [Conitzer, 2006]. However, because the constraints are all relaxed in this setting, and because payments can be non-monotonic in value, we obtain less power by pruning the formulation in this way, as can be obtained in AMD.

3.5.3.3 Core

In settings, such as combinatorial auctions, where we may desire stability against the threat of sets of agents breaking away from the mechanism, we may consequently want to enforce core constraints. Equation (3.36) is a γ-relaxed version of such constraints, enabling it to find *Least-Core* prices that approximate nucleolus prices without requiring lexicographic search (see 2.9.2, [Yokoo et al., 2005]). Because the relaxation enables us to leave the core itself, we can find payments that are applicable to combinatorial exchange settings, where the core may be empty. Following the method proposed by Day and Cramton [2008], we enforce that the payments made by winning agents outside the coalition must be greater than the additional amount that agents inside the coalition could make by breaking off on their own. This condition is enforced for all bid profiles being considered, and all agent coalitions.

3.5.3.4 Balance

The objective defined so far is not sufficient to ensure a unique payment vector. We therefore break ties by choosing prices that are as similar as possible. Specifically, we minimize the distance between the payments and a centroid point ζ, as specified in equation (3.37). Notice that this definition only applies to the price vector corresponding to the reported bins $\hat{\mathbf{v}}$, as we only require uniqueness in the vector we are actually going to implement.

3.5.3.5 Constraints

We need to ensure that the various prices are Individually Rational. We therefore include equation (3.38), which ensures that all of the prices specified are weakly less then the corresponding representative value for the associated bin. Additionally, we ensure that price vectors associated with each hypothetical report profile are budget balanced in equation (3.39), and specify the variable domains in equation (3.40).

3.5.4 Alternative Formulations

In addition to the *ex ante* formulation just given, we can also define the other objectives described in Section 3.3. For example, equation (3.32) specifies the expected gain of each agent. If we replace the summation with a maximization, we obtain the worst case instead.

3.5.4.1 Percentile

We can target a given percentile of the gain distribution instead of the expected case analysis above. This requires converting our linear program into a mixed integer program by replacing equation (3.32) with:

$$\delta_i = P_\rho \left(\{ \langle f_{\mathtt{v}_i}(\mathtt{v}_i), \delta_{i,\mathtt{v}_i} \rangle : \mathtt{v}_i \in \mathtt{V}_i \} \right) \tag{3.42}$$

Where $P_\rho(\cdot)$ represents the ρ percentile of the given probability-weighted set of variables, as defined below.

3.5.4.2 Formulating Percentiles in a MIP

To make this conversion, we need a mixed integer formulation for calculating the ρ-th percentile (expressed $\rho \in [0, 1]$) of a set of probability-weighted program variables x_k with corresponding probabilities ω_k. Specifically, we provide a formulation for $x_\rho = F_Y^{-1}(\rho)$, where the random variable $Y = \vec{\omega} * \vec{x}$ and $*$ is the element-wise product of two vectors. That is, we calculate the inverse CDF or quantile function of the distribution of Y.

We introduce $|K|$ binary variables β_k, that specify which variables in K are less then the given percentile. To define these variables we enforce two *big-M* constraints per variable in K:

$$M_k \beta_k + x_k \leq M_k + x_\rho \;\forall\; k \in K \tag{3.43}$$

where $M_k = \overline{x}_k - \min_{j \in k} \underline{x}_j$. And:

$$M_k \beta_k + x_k \geq x_\rho \;\forall\; k \in K \tag{3.44}$$

where $M_k = \max_{j \in k} \overline{x}_j - \underline{x}_k$.

With the β_k variables defined, we can ensure that the variable x_ρ will correspond to (approximately) the desired percentile, by enforcing:

$$\sum_{k \in K} \omega_k \beta_k \geq \rho \tag{3.45}$$

Here we assume the calculated percentile variable x_ρ will be minimized; if it will be maximized, then we must reverse the inequality in (3.45). Note that in this formulation we have not attempted to interpolate between the data points, which would increase fidelity when $|K|$ is small.

3.5.4.3 Ex Post

In the formulation given above the determination of the optimal deviation is made with respect only to expected information about the bids of the other agents. As such it represents an *ex ante* optimization as defined in Section 3.3.2.1. If we swap the chaining in equations (3.33) and (3.34), we will obtain an *ex post* optimization as in Section 3.3.2.4.

3.5.5 Extensions

In the above formulation we define joint, albeit bucketed, bid profiles across all the agents. For large numbers of agents and complex bids, this is very expensive. To decrease the formulation size we can make a further approximation in the joint bid space, an extension we leave for future work. Specifically, instead of a full Cartesian product of individual agent bid profiles, one could reduce the number of trades under consideration in the set $\mathtt{V}_{\text{-}i}$. That is, use a large number of trades only for agent i, while only a small number of trades (and possibly only the implemented trade) for all agents other than i.

3.6 Example 1: Two Buyers, One Good

To make the formulation presented in Section 3.5 clear, we illustrate it with a simple example. Suppose we have a basic auction with a no-reserve seller offering a single item and two buyers. Further, we take the prior distribution on buyer value for the item as uniform $[0, 1]$. Now consider a particular market instance where: agent 1

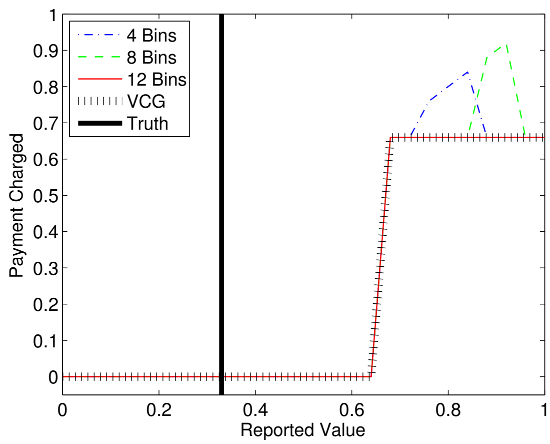


Figure 3.1: Agent 1 payments by report and bin

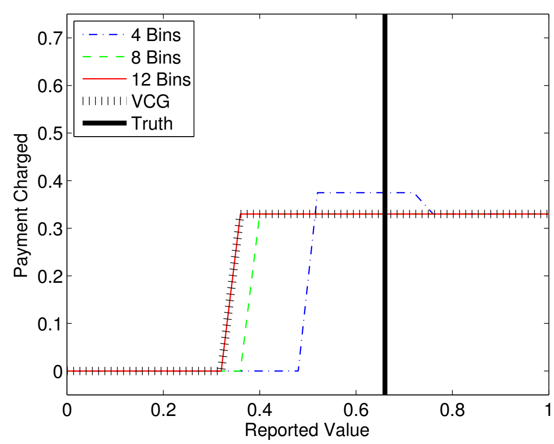


Figure 3.2: Agent 2 payments by report and bin

has a true value of 1/3, and agent 2 has a true value of 2/3. In this simple setting the efficient trade is for agent 2 to win, and VCG would charge a payment from agent 2 equal to the bid of agent 1.

3.6.1 Actual Payments

With this setup, we can consider the profile of payments charged to each agent as they vary reports. Figures 3.1 and 3.2 show the payments charged to agent 1 and 2 respectively, when the expected case *ex ante* LP is formulated with 4, 8 and 12 bins. The LP is run 'on the fly' for different inputs, each input corresponding to one point on the 'reported value' axis of the graph. The VCG payments are included for comparison purposes.

With very small numbers of bins there is some fidelity loss, and for some reports the agents may indeed have an incentive to deviate. However as the number of bins becomes large enough (i.e. 12), the formulation finds the optimal solution and

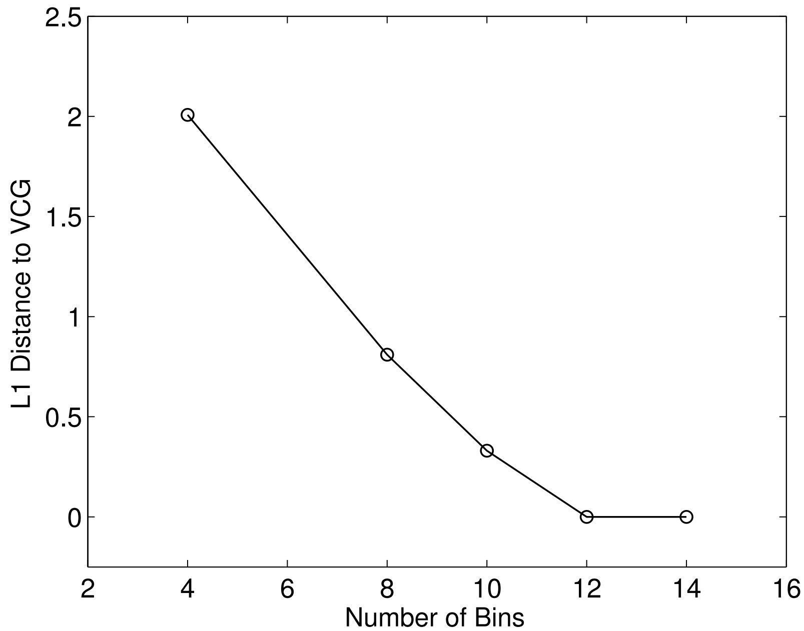


Figure 3.3: L_1 distance to VCG payments by number of bins

recreates the VCG payments exactly.

Thus, the example allows for a direct examination of the relationship between the number of bins and the fidelity of the induced approximation, because it admits a fully strategyproof mechanism, namely VCG. To illustrate this, we can draw 100 value profiles from our setup (i.e. 100 buyer pairs, each drawn $U[0, 1]$). For each of these instances, we then solve for both the VCG payments and the automated payments with various numbers of bins. We can then calculate the L_1 distance between the automated and VCG payments. This is plotted in Figure 3.3 as a function of the number of bins in the approximation.

We can see that the fidelity is improving rapidly, and that ultimately exactly matching payments are found with only a relatively modest number of bins. In more complex scenarios the VCG solution is not admissible (because it's e.g. out of the

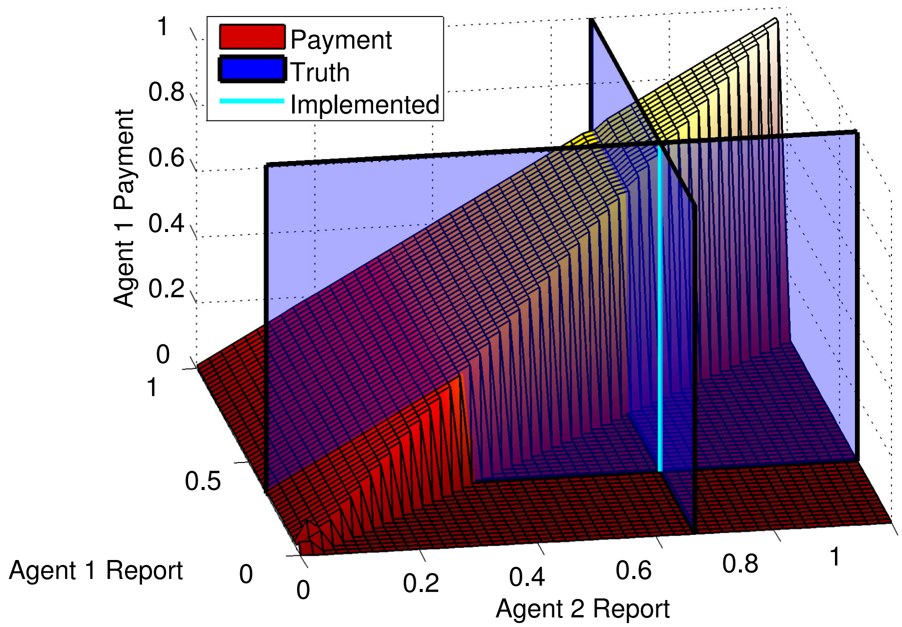


Figure 3.4: Counter-factual payments for agent 1

core, or not budget balanced), so we won't have a gold standard available for such direct comparisons, but a modest number of bins is still likely to obtain a result of reasonable fidelity.

3.6.2 Counter-factual Payments

It is important to realize that the formulation calculates payment vectors not only at the reported values, but also for many other high-probability bids according to the prior distribution. In this simple domain, we can show these counter-factual payments directly. Figures 3.4 and 3.5 show the counter-factual payment calculated for each agent as a function of the counter-factual joint report of the agents. These are payments calculated by the LP specifically formulated for the given reports, $(1/3, 2/3)$, shown by the translucent planes in the figure. The line at the planes' intersection

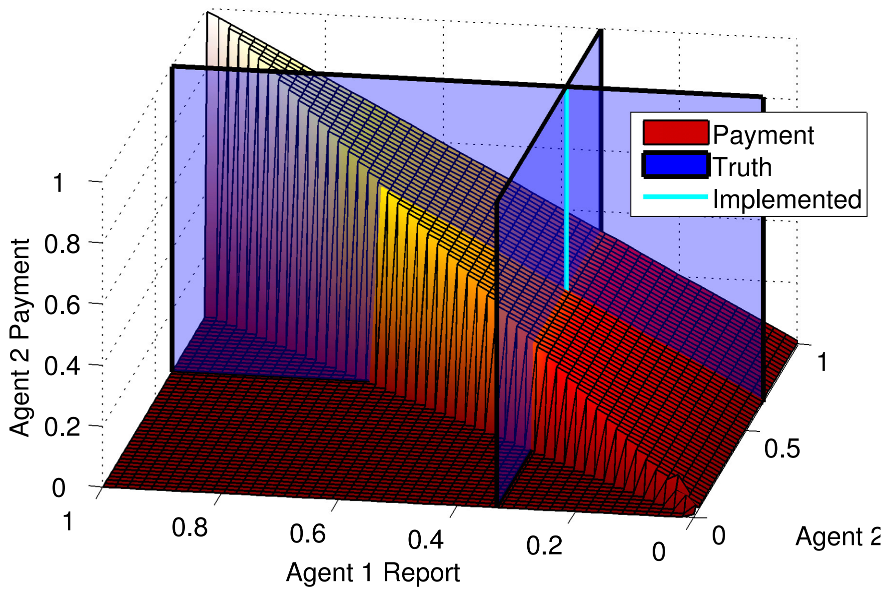


Figure 3.5: Counter-factual payments for agent 2

corresponds to the payment that is actually chosen.

The graph shows the scenario exactly as above, but with 50 bins per agent to increase the graph resolution. A single payment variable is simultaneously calculated for each non-zero intersection of the grid shown. It is worth noting that in the dark triangular area with zero height there are no corresponding payment variables in the formulation. This is because the agent's reported values in that range are too small to win the trade being considered; and as this is known a priori, these payments are known to be zero without the need to explicitly include variables in the formulation. In this example, this observation makes the LP formulation about half the size it might otherwise be – and in general the optimization significantly reduces the size of the formulation.

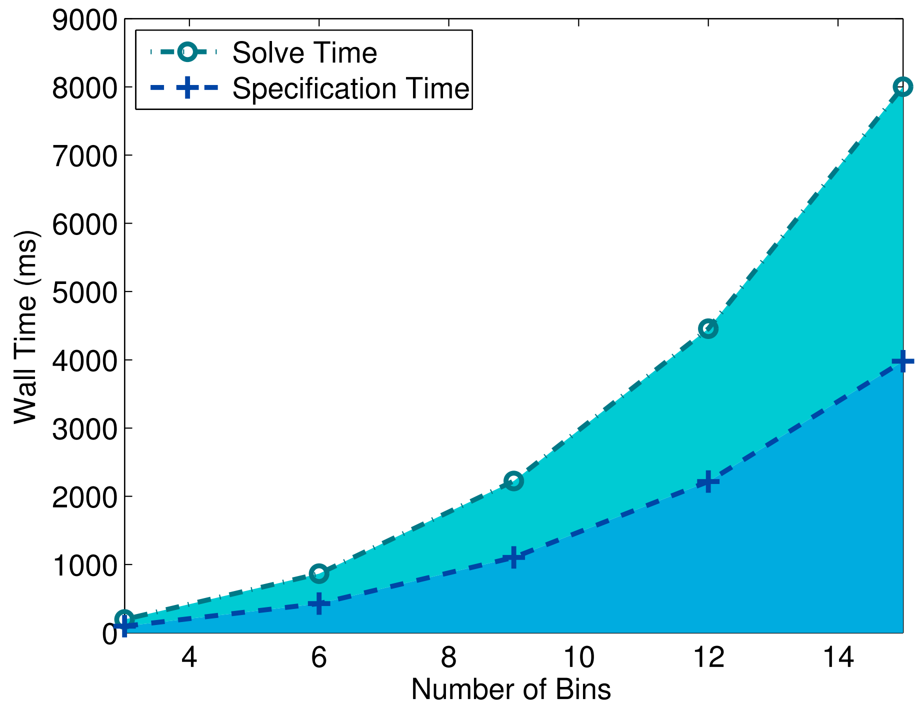


Figure 3.6: Scalability by number of bins

3.6.3 Scalability in Number of Bins

The formulation is exponential in the number of bins, so we expect computational time to increase accordingly. This is confirmed by the graph in Figure 3.6 which shows computational time on a dual quad-core 2.83GHz Xeon workstation with 8GB of memory. The time shown is the time needed to specify the LP to ILOG CPLEX 11.1 via its Java APIs, and then the time needed to solve the LP. While it is a multi-core machine, CPLEX was not configured to run extra threads, so this represents serial computational time. Note that the considerable time needed to compute the probability distribution by sampling to reasonable fidelity is performed in parallel and is not included in the graph. It is assumed that in any practical use, such a prior would be available directly from historical data, and so such data would not need to be generated on the fly. As is evident from the graph, simply specifying the large LPs involved represents significant computational burden. If we seek a solution that is best in either the expected or worst cases, we end up with an LP formulation (as shown in the graph). If instead we seek the solution at a given percentile we end up with a MIP. Solving an LP is in theory polynomial in its size, but our size is growing exponentially in the number of bins, so the solve time grows exponentially in the number of bins even when we use an LP solution. Thus going to the MIP formulation is necessary, for the percentile formulation may not represent a huge additional burden in practice.

3.7 Example 2: Three Buyers, Two Goods

Next we consider the more complex domain examined by Erdil and Klemperer in their recent economics theory paper [2010]: a combinatorial auction of two goods, A and B. In addition to the no-reserve seller, we have three buyers; buyer 1 desires good A, buyer 2 desires good B and buyer 3 desires the bundle AB. Again we assume agent's values are drawn from $U[0, 1]$, and that we have access to an accurate prior. In a given instance, we'll refer to the buyer's bid vector as (v_1, v_2, v_3).

3.7.1 Other Rules in this Setting

In order to evaluate our automated design rule in this expanded setting, we benchmark it against several other rules from the literature. All of the following rules are identical when $v_3 > v_1 + v_2$: buyer 3 wins the bundle and pays $v_1 + v_2$. The interesting case is when buyers 1 and 2 win:

3.7.1.1 Reference

Erdil and Klemperer [2010] propose a *Reference* rule specifically for (and their analysis is limited to) this two-goods auction setting. They propose choosing a reference point $(r_1, r2)$. Then, in the case where buyers 1 and 2 win, their rule stipulates that buyer 1 should pay $min(v_1, r_1)$ and that buyer 2 should pay $min(v_2, r_2)$. The exact choice of the reference point is left open, but $v_3/2$ is suggested, and is thus used here.

3.7.1.2 Euclidean

An alternative is to minimize the Euclidean distance to VCG. In this setting this reduces to the following rule when buyers 1 and 2 win: define a discount budget as $\beta = v_1 + v_2 - v_3$. Then charge agent 1 $v_1 - \beta/2$ and agent 2 $v_2 - \beta/2$. In this two-goods setting, several rules from the literature collapse to this rule. In particular the Threshold, Reverse, Fractional, Equal rules proposed Parkes et al. [2001b] and described in Section 3.2.1 are in this class.

3.7.1.3 Extreme

Section 3.2.1 also described the Small and Large rules defined by Parkes et al. [2001b]. In this setting and when buyers 1 and 2 win, these rules collapse to the following:

$$\begin{array}{rl} \text{With probability } {}^1\!/_2\text{:} & \text{Buyer 1 pays } v_3 - v_2, \\ & \text{Buyer 2 pays } v_2 \\ \text{And probability } {}^1\!/_2\text{:} & \text{Buyer 1 pays } v_1, \\ & \text{Buyer 2 pays } v_3 - v_2 \end{array}$$

3.7.1.4 Balance

We also consider a rule that attempts to 'balance' the payments. That is, it attempts to equalize the payments (and not the payoffs) while respecting the budget balance, individual rationality and core constraints. Specifically:

$$\begin{array}{lll} \text{If } v_1 \geq {}^{v_3}\!/_2 \text{ and } v_2 \geq {}^{v_3}\!/_2\text{:} & & \text{Buyers 1 and 2 both pay } {}^{v_3}\!/_2 \\ \text{Otherwise:} & \text{If } v_1 > v_2 : & \text{Buyer 1 pays } v_3 - v_2, \\ & & \text{Buyer 2 pays } v_2 \\ & \text{Otherwise:} & \text{Buyer 1 pays } v_1, \\ & & \text{Buyer 2 pays } v_3 - v_1 \end{array}$$

3.7.2 An *Ex Post* Analysis

We would now like to compare our Automated rule to these various alternatives within this small CA domain. Interestingly, we now have a question as to what the best way of making such a comparison might be. We might choose to calculate the full BNE under each rule, and make a comparison that way. But this is extremely computationally expensive, especially since the Automated rule must solve an expensive optimization, unlike more typical closed-form rules. We therefore reserve performing a full equilibrium analysis until Chapter 4, where we will approximate a BNE in an even more complex setting (but without rules as complex as Automated).

So instead, at the risk of a circularity of argument, we might choose to evaluate the Automated rule according to any of the same criteria defined in Section 3.3 as our means for making the comparison. There is reason to like the expected-case *ex ante* criterion as structurally closest to a full BNE. However, its calculation is also difficult, as the calculation of optimal best responses under the *ex ante* information regimen require solving a very large stochastic optimization problem. We thus choose the quantile *ex post* criteria, as it is computationally tractable, and as we shall see, highly informative of the incentive structure of a mechanism.

This choice will mean that agents have full information when deciding whether and how to misreport, and in particular they know the other agents' valuations. As such, it is a pessimistic analysis in the sense that agents will often have far less information in practice, and thus they will not be able to manipulate as effectively.

In order to study the distribution of quantiles of the *ex post* gain experimentally, we need a way to determine *ex post* best responses in a given market instance. We can find this quantity by first replacing the agent's bid by a single-minded bid for his winning trade. Then, we search across all possible value reports the agent might make on this trade, while holding all the other agents constant. In the experiment presented here, this linear search operation is accomplished with a Brent maximization [2002], set to find a solution within 5% of optimal.

By sampling 100 market instances we obtain a collection of possible gains from deviation. One way to evaluate these is to examine the maximum gain over the sample set (i.e. 100^{th} percentile). Parkes' et. al. Threshold rule [2001b] (i.e. the *Euclidean* rule in this setting) is known to minimize the maximum gain, and thus will have the

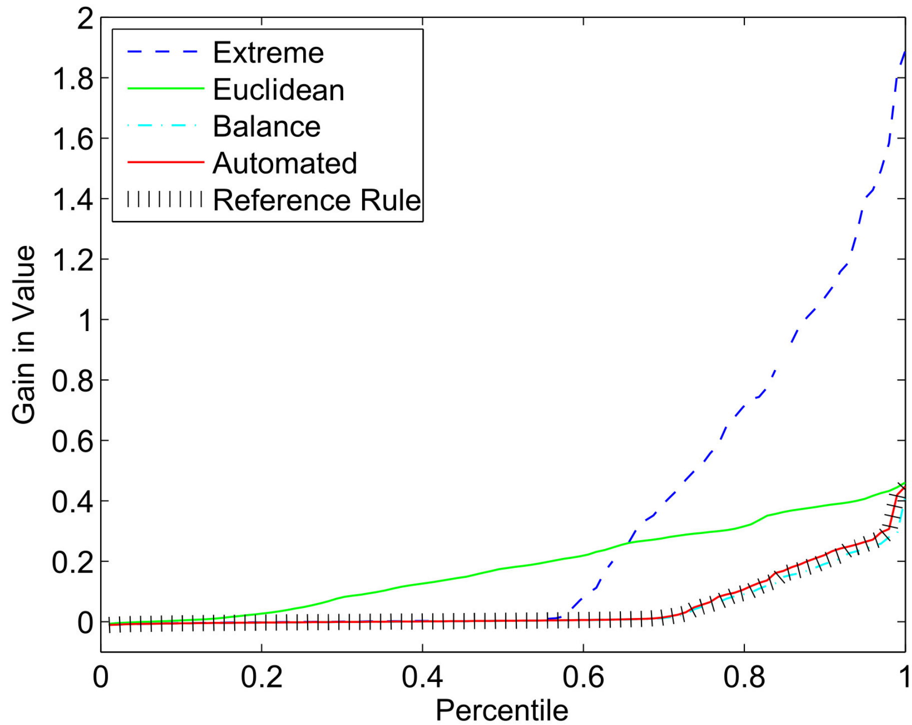


Figure 3.7: Percentile of *ex post* gain available under each rule

smallest gain at the 100$^{\text{th}}$ percentile for any reasonable sample size. While consistent with the worst case analysis typically considered in prior approximate-strategyproof analysis, this view does not consider any of the instances other than the very worst one. Instead, we might consider the mean gain under a given rule, but nuances in the behavior outside of the worst case are made even more manifest by considering the quantiles of the gain distribution instead (as suggested in Section 3.3.2.3). The quantiles of a given distribution are the values of the inverse cumulative distribution function.

Accordingly, Figure 3.7 shows the percentiles of the unilateral gain distribution under each of the rules described above as well as the *Automated* rule evaluated at the expected case *ex ante* objective and 10 discrete bins The percentile plot is constructed by graphing a Gaussian kernel-smoothed version of the empirical inverse cumulative distribution function. The kernel smoothing helps to increase the fidelity of the graph where there are discontinuities in the empirical CDF due to rare events and thus low data density.

The graph shows the strong performance of the Erdil and Klemperer *Reference* rule. This is perhaps unsurprising as it was developed explicitly for its performance specifically in this domain and with this prior. More interesting, however, is that nearly identical performance of the *Balance* and *Automated* rules despite the approximations in the latter. In this setting, the *Extreme* rules are clearly doing very poorly in comparison to the other rules. As expected the *Euclidean* (or Threshold) rule is optimal at the worst case (100^{th} percentile), but in this setting, it is inferior everywhere else. Thus, contrary to earlier work, this argues that looking at the worst case may not be the best criterion for either designing or choosing a rule, a topic we return to in Chapter 4.

3.7.3 Distance from Reference

While Figure 3.7 shows the the near identical performance of *Reference*, *Balance* and *Automated*, it does not provide insight as to the similarity of the payments actually charged. As it turns out these three rules choose nearly identical payments in this setting, as indicated by the L_1 distances from payments charged by each of the rules to that charged by the *reference* rule provided in table 3.1.

Rule	$L_1/\vert N\vert$
Balance	0.0010
WC Automated *Ex Post*	0.0042
WC Automated *Ex Ante*	0.0048
Euclidean	0.1605
Extreme	0.3023

Table 3.1: L_1 distance to the reference rule

The table shows the distance for the *Automated* rule under both an expected *ex ante* and expected *ex post* criterion are very similar. Thus in this setting probabilistic information about other agents valuations is sufficient to design a highly effective rule. However, this similarity is unlikely to remain in more complex settings.

3.8 Observations

We have examined ways to formalize what we mean by "approximately strategyproof" in over-constrained mechanism design settings. In particular, we have identified two orthogonal criteria to consider in such a definition:

- **Information available to agents:** Agents that are fully informed about the market in which they are bidding, and in particular about the values of the other bidders, are in a very strong position when behaving strategically. A mechanism resilient to these types of agents has *ex post* approximate incentive compatibility. By contrast, agents that have only probabilistic information about other agents bids are in a weaker position. A mechanism resilient to these types of agents has *ex ante* approximate incentive compatibility, which is characterized by tending to a Bayes-Nash equilibrium, as the approximation goes to zero and in the special case where the value distributions are drawn from equilibrium data. Despite additional computational requirements *ex ante* quantities are likely to be relevant because agents are often poorly informed.

When using our criteria constructively, as we do in this chapter, there is little loss in using an *ex ante* mode if one believes agents to be poorly informed – the computation is expensive either way. However, if we are using the criteria for evaluating existing rules, we must calculate best responses. As calculating *ex ante* best responses requires stochastic optimization this is far more expensive than an *ex post* analysis, and in this case, we therefore argue for using the *ex post* criterion together with quantiles.

- **Worst/Expected Case:** Do we consider the maximum gain from deviation that is possible across all possible agent values? Or, do we consider the expected gain from deviation under our prior over agent values? A risk neutral mechanism designer may choose to sacrifice the worst case in order to have a lower expected gain. We also propose considering a quantile (e.g. 75^{th} percentile) of gain from deviation instead. Such a choice balances the desire to have good expected case behavior, against robustness under the most extreme value profiles. We show how to formulate such an objective in an *Automated* rule search.

  While all three approaches are valid, we belive the quantile aproach holds tremendous value. It enables examination of the tradeoff between good behavior in the common case, and robustness to bad instances. Moreover, when forced to use an *ex post* critera for computational reasons when performing a comparison between rules, the additional power of the quantile analysis enables considerable nuance in understanding the incentive properties of a mechanism by relaxing away from the absolute worst case that one obtains under a regret critera in an extremely useful way.

Having defined these objectives, we can formulate the choice of maximally incentive-compatible payment rules as an optimization problem. This optimization problem is related to the Automated Mechanism Design work of Conitzer and Sandholm (See Section 4.9.3.2). However, it differs in significant ways. We search for the most strategyproof mechanism subject to design constraints (e.g. Budget Balance, Individual Rationality, Core, etc.), instead of searching for the design that leads to the highest social welfare, subject to a strategyproofness constraint. We are thus solving a fundamentally different constrained optimization problem. Secondly, we do not optimize over the outcome trade selection, instead preferring to hold this fixed, e.g. at the efficient trade for the reported values.

While we have written down a theoretically optimal constrained optimization problem according to our chosen approximate incentive compatibility criterion, it was an optimization over a space of functions, and not scalars. There are no known techniques for analytically solving such complex functional optimization problems. So instead we proposed to approximate the problem using standard linear or mixed integer programming. The practical formulation of such approximation programs requires special techniques, including inverse transform sampling for value discretization and various approximations notably the careful selection of which trades to consider.

We have shown that these approximations can yield good results – recreating known optimal rules in the simple domains for which they are tractable. However, even with the approximations proposed, solving for the approximately optimal rule is expensive, as we saw in Section 3.6.3. So rather then explicitly solving for the optimal mechanism, we may instead wish to use closed form rules that are inexpensive to calculate; but to do so we need a way to compare proposed mechanisms with respect to their approximate incentive compatibility that is also inexpensive. This will enable us to consider far more complex domains including full combinatorial exchanges, as we will see next.

Uniformity in [the] currency, weights, and measures [of the United States] is an object of great importance, and will, I am persuaded, be duly attended to.

– George Washington
First State of the Union Address, 1790

4

Quantifying the Incentive Properties of Payments

4.1 Motivation

As we have seen in chapter 3, formulating and solving the optimization problem necessary to find approximately incentive compatible payment rules is computationally expensive because formulations tend to grow exponentially in the number of agents. So for reasonably sized problems in such settings we will want to use an easy to calculate closed-form payment rule that is not necessarily strategyproof. Accordingly, in order to select among several such rules, we seek a method for rule comparison according to approximate incentive compatibility.

Thus, we introduce as a metric the *normalized Kullback-Lieber (KL) Divergence* between the distribution of payoffs in a mechanism and a distribution induced by a strategyproof "reference" mechanism, where these payoffs (or utilities) are evaluated given truthful bids (i.e. out of equilibrium), and are restricted to agents affected by the outcome (either positively or negatively.)[1] Consequently, the metric requires that there exist a strategyproof reference mechanism for some natural relaxation of the problem. The motivating hypothesis is that the closer a given mechanism is to the strategyproof reference *in distribution*, the more strategyproof it will be. We contrast this approach with earlier work that has focused on *per instance* measures of strategyproofness. This approach of evaluating already-defined rules also stands in contrast to the constructive, and thus expensive, approach we took in chapter 3.

The particular metric that we develop is applicable to general mechanisms which select outcomes separately from the payments that they charge. This enables us to hold the outcome rule constant between the reference and the mechanism being evaluated, and concentrate solely on the payment rule. Combinatorial Exchanges are a motivating member of this class, and the one focused on throughout this chapter.[2] For such a setting, the Vickrey-Clarke-Groves (VCG) mechanism is revenue-maximizing amongst individually rational, strategyproof and efficient mechanisms, as discussed in Section 2.7.2 [Holmström, 1979]. While it is strategyproof and efficient, it typically runs at a deficit precluding its direct use – but it is an ideal reference mechanism on

[1] Throughout this chapter, we for simplicity refer to the KL-divergence as a metric. However, by a strict mathematical definition this measure of distance between two distribution functions is a *quasi-metric*, not a true *metric* because it is not symmetric. This lack of symmetry has no consequence for our use.

[2] While we focus on budget balanced combinatorial exchanges in this chapter, the other settings discussed in chapter 3 are also applicable to this methodology. These include core-constrained combinatorial auctions [Ausubel and Milgrom, 2006], and mechanisms constrained by their need for simplicity, such as sponsored search generalized second price (GSP) auctions [Lahaie et al., 2006].

which to use our metric.

In this chapter we evaluate the KL-divergence, and a number of other regret-based metrics, on a family of approximately strategyproof mechanisms that were proposed in Parkes et al. [2001a] and described in Section 3.2.1. In providing validation results, we need to adopt an approximate method to compute equilibrium in different CE mechanisms because there is no computationally tractable method to compute exact Bayesian-Nash equilibrium in CEs. For this, we compute restricted, partially-symmetric equilibria.

As we shall see, the KL-divergence metric has a significant and strongly positive correlation with a parametrization of the amount by which the equilibrium deviates from truthful reports (in approximate Bayes-Nash equilibrium), and a strongly negative correlation with the allocative efficiency in equilibrium. The metric identifies the Small rule from Parkes et al. [2001a] as the best mechanism, and it is indeed this rule that provides highest efficiency and least bid-shaving in equilibrium. In testing the power of the metric for mechanism design, we show that it is effective in guiding a search through a set of mechanisms and identifying a highly efficient mechanism based only on observed data.

4.1.1 A Heuristic Mechanism Design Paradigm

By way of motivating the need for a metric to quantify approximate strategyproofness, consider the following heuristic approach to mechanism design:[3] there is a space of non-strategyproof mechanisms $\mathcal{M}$, each of which has the same outcome rule and

[3] While the form being advanced here is distinct, other approaches to heuristically designing mechanisms were advocated by Parkes [2009].

good properties when agents are truthful, and with properties that degrade as agents becomes less truthful in equilibrium. *Given this set of mechanisms, adopt as the goal that of selecting the mechanism in $\mathcal{M}$ that is maximally strategyproof.* For example, these could be mechanisms in which outcome rule $\mathcal{W}(\boldsymbol{v}) \in \arg\max_{a \in A} \sum_i v_i(a)$ but vary in their payment rules, so that if agents are truthful the mechanism is efficient; i.e., maximizing the total value through its choice of alternative. In doing so, we seek a metric on approximate strategyproofness that provides explicit design guidance because the space of mechanisms may be too large to enumerate, and that works without requiring the computing of the equilibrium of a candidate mechanism because this is computationally expensive.

A standard answer would be to select a mechanism that minimizes the worst-case *ex post* regret from behaving truthfully, across all agents and across all instances [Schummer, 2001]. The regret of agent i when valuations are $\boldsymbol{v} = (v_1, \ldots, v_n)$ is $regret_i(\boldsymbol{v}) = \max_{\hat{v}_i}[v_i(\mathcal{W}(\hat{v}_i; \boldsymbol{v}_{\text{-}i})) - p_i(\hat{v}_i; \boldsymbol{v}_{\text{-}i})] - (v_i(\mathcal{W}(\boldsymbol{v})) - p_i(\boldsymbol{v}))$. Complexity aside, we can also question whether this is the right answer. Does this lead to a mechanism in which an agent's equilibrium bids are closer to truthful, on average, than in the other mechanisms in $\mathcal{M}$?

In the formalism we introduced in Section 3.3, maximal regret measures the absolute worst case *ex post* gain. In chapter 3 we showed for simple domains that by focusing on the worst case behavior we potentially traded away the opportunity for decreasing the incentive to misreport in the rest of the cases. The logical question to ask is if this behavior extends to the more complex combinatorial exchange setting considered here.

4.2 The Setup

4.2.1 The Design Space

The approximately strategyproof payment rules proposed by Parkes et al. [2001a] and described in Section 3.2.1 will play the role of the design space $\mathcal{M}$ in this chapter (with one addition, to be described shortly). Each mechanism adopts the same allocation rule as in VCG (and therefore has good properties when agents are truthful) but defines payments that are exactly balanced. Conceptually, the payment rules all discount the amount an agent i will pay relative to its reported valuation $\hat{v}_i(\lambda^*)$ for the selected trade. In the VCG mechanism, this discount is $\Delta_{\text{vcg},i}(\hat{\boldsymbol{v}}) = V^*(\hat{\boldsymbol{v}}) - V^*(\hat{\boldsymbol{v}}_{-i})$, but in each of these new mechanisms the discounts are constrained so that $\sum_i \Delta_i(\hat{\boldsymbol{v}}) = V^*(\hat{\boldsymbol{v}})$, providing $\sum_i p_i(\hat{\boldsymbol{v}}) = \sum_i(\hat{v}_i(\lambda^*) - \Delta_i(\hat{\boldsymbol{v}})) = V^*(\hat{\boldsymbol{v}}) - V^*(\hat{\boldsymbol{v}}) = 0$ and no-deficit, where $V^* = \sum_i v_i(\mathcal{W}_i(\boldsymbol{v}))$ is the total surplus generated at the optimal trade. The deviation from the payments of the VCG mechanism opens up the possibility that an agent can gain by deviating from its truthful report. Each mechanism in $\mathcal{M}$ adopts a different method to allocate the available surplus to agents. The regret of agent i is exactly $\mathit{regret}_i(\hat{\boldsymbol{v}}) = \Delta_{\text{vcg},i} - \Delta_i(\hat{\boldsymbol{v}})$, i.e. the amount by which the discount is less than that in the VCG mechanism. The Threshold rule has been considered of particular interest because it defines payments that minimize the maximal regret to agents, given the no-deficit constraint.

In addition to the rules described in Section 3.2.1 we also introduce a new rule: the "TwoTriangle" rule allocates half of the surplus by Threshold and then overlays the remainder by Small. This results in a rule that performs nearly as well as Small,

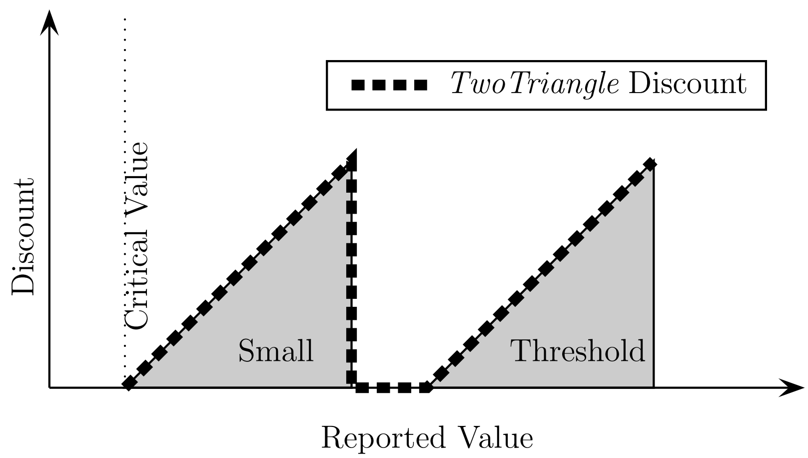


Figure 4.1: Stylized sketch of the discounts provided by the *TwoTriangle* rule, as a function of reported value. Note that this picture treats the total available surplus as constant, when it in fact will change, as it is dependent on the report. While this effect can be important in practice, the sketch gives a good intuition for the behavior of the rule.

but without the undesirable property that agents with the largest discounts receive no profit at all, unless they are untruthful. The name derives from the tendency of the rule to exhibit a discount shape characterized in Figure 4.1. When the report is large relative to the critical value, the bid will fall in the triangular area on the right, and have a discount provided by the Threshold rule. When the report is small relative to the critical value, the bid will fall in the area to the left and have a discount provided by the small rule. Depending on the specifics of the instance, agents in the middle may receive a modest discount, or none at all. The version of TwoTriangle evaluated here splits the available surplus to be distributed 50%-50% between Small and Threshold. The amount given to each rule to allocate could be assigned in any other proportion according to the designers' choice in the trade-off between the good equilibrium incentive properties of Small and the more desirable treatment of agents with large discounts exhibited by Threshold.

4.2.2 The Multi-Dimensional KL-Divergence

Given the complexity of the approaches in the previous chapter, here we propose a metric that adopts a strategyproof *reference mechanism* m^*, and seeks a mechanism m that induces payoffs that are close in distribution to m^*. The reference mechanism will be outside of $\mathcal{M}$ (the set of mechanisms meeting our design objectives), and with the same outcome rule but a payment rule that makes the mechanism strategyproof. A good reference mechanism should be strategyproof and as close to satisfying the properties of $\mathcal{M}$ as possible.

For a particular instance, let $\pi^m(\boldsymbol{v}) = (\pi_1(\boldsymbol{v}), \ldots, \pi_n(\boldsymbol{v}))$ define the payoff to each agent in m, i.e. $\pi_i(\boldsymbol{v}) = v_i(\mathcal{W}(\boldsymbol{v})) - p_i(\boldsymbol{v})$. Similarly, let $\pi^*(\boldsymbol{v}) = (\pi_1^*(\boldsymbol{v}), \ldots, \pi_n^*(\boldsymbol{v}))$ define the payoff to each agent in the reference mechanism m^*. Let $\pi \in \Pi$ be a feasible joint payoff vector and let $H^m(\pi)$, $H^*(\pi)$ be the joint distribution of payoffs under mechanism m and m^* respectively, as induced by a distribution on valuations. Then consider the full multivariate KL-divergence between these distributions: $\int_{\pi\in\Pi} H^*(\pi) \log(\frac{H^*(\pi)}{H^m(\pi)}) d\pi$.

However such an approach is complex, and requires a lot of data to have sufficient density in the empirical joint distribution space. So, to keep things relatively simple, we will consider in this paper a projection of these multi-dimensional distributions down to one-dimensional payoff distributions. To do this, one could obtain a single-variate metric based on the payoff distribution of each agent independently, and then take a summary statistic over these agent-specific measures. But this introduces an extra layer of indirection between our final metric and the multi-variate distribution. So instead, we choose to simply combine the single agent payoff distributions, and then compute a single univariate metric across all of the agents at once. This is effectively a projection in the probability space, and we have found it to be effective.

4.2.3 Normalized KL-Divergence

In the CE environment, we specialize the general multivariate KL-divergence to a KL-divergence on *normalized* payoff, where the payoff $\pi_i^m(\boldsymbol{v})$ to each agent in instance $\boldsymbol{v}$ is normalized by $V^*(\boldsymbol{v})$. Specifically, the normalized KL-divergence for mechanism m is defined as:

$$KLnorm(m) = \int_0^\infty \widehat{H}^*(\pi) log \left(\frac{\widehat{H}^*(\pi)}{\widehat{H}^m(\pi)} \right) d\pi, \tag{4.1}$$

where $\widehat{H}^*(\pi)$ is the univariate distribution of the normalized payoff $\frac{\pi_i^*(\boldsymbol{v})}{V^*(\boldsymbol{v})}$ under the reference mechanism, given the distribution on instances, and $\widehat{H}^m(\pi)$ is similarly defined for the mechanism being considered. We further restrict these distributions to payoffs associated with agents that are active in the efficient trade. Note that the distribution on payoffs is that induced by the *true* distribution on valuations, not by the equilibrium distribution. We also consider an unnormalized KL-divergence.

Special care must be taken in applying the KL-divergence to settings where the distributions that it is measuring may place zero probability mass on certain outcomes, as this can either result in a $log(0)$ or a division by zero. This can pose a particular experimental problem in that certain portions of the support of the distribution may not have received any samples, even when the true distribution places mass at these locations. This can be compounded by numerical integration methods used to compute Equation (4.1). To overcome these problems we generate empirical

PDFs that have full support by using a Gaussian kernel to smooth the density of raw samples we have obtained.

4.2.4 Regret Based Metrics

In addition, we study a number of traditional, regret-based, metrics:

$$L_1(m) = \int_v ||\pi^*_+(\boldsymbol{v}), \pi^m_+(\boldsymbol{v})||_1 f_{\boldsymbol{v}}(\boldsymbol{v})dv \tag{4.2}$$

$$L_1\, norm(m) = \int_v ||\frac{\pi^*_+(\boldsymbol{v})}{V^*(\boldsymbol{v})}, \frac{\pi^m_+(\boldsymbol{v})}{V^*(\boldsymbol{v})}||_1 f_{\boldsymbol{v}}(\boldsymbol{v})dv \tag{4.3}$$

$$L_2(m) = \int_v ||\pi^*_+(\boldsymbol{v}), \pi^m_+(\boldsymbol{v})||_2 f_{\boldsymbol{v}}(\boldsymbol{v})dv \tag{4.4}$$

$$L_2\, norm(m) = \int_v ||\frac{\pi^*_+(\boldsymbol{v})}{V^*(\boldsymbol{v})}, \frac{\pi^m_+(\boldsymbol{v})}{V^*(\boldsymbol{v})}||_2 f_{\boldsymbol{v}}(\boldsymbol{v})dv \tag{4.5}$$

$$L_\infty(m) = \int_v ||\pi^*_+(\boldsymbol{v}), \pi^m_+(\boldsymbol{v})||_\infty f_{\boldsymbol{v}}(\boldsymbol{v})dv \tag{4.6}$$

$$L_\infty\, norm(m) = \int_v ||\frac{\pi^*_+(\boldsymbol{v})}{V^*(\boldsymbol{v})}, \frac{\pi^m_+(\boldsymbol{v})}{V^*(\boldsymbol{v})}||_\infty f_{\boldsymbol{v}}(\boldsymbol{v})dv \tag{4.7}$$

where $f_{\boldsymbol{v}}(\boldsymbol{v})$ is the PDF on valuation instances $\boldsymbol{v}$ (for the truthful distribution), $\pi^*_+(\boldsymbol{v})$ and $\pi^m_+(\boldsymbol{v})$ indicate the payoff vectors restricted to agents that are active in the trade, and $L_1(\cdot,\cdot), L_2(\cdot,\cdot), L_\infty(\cdot,\cdot)$ are standard L_1, L_2 and L_∞ metrics. These metrics are evaluated on a per-instance basis. The integration above simply evaluates the raw metric over multiple (and at the limit, an infinite number of) sample instances. So, although all of the metrics above are defined over a continuous valuation space, practical evaluation will require numerical integration over specific samples.

The L_1 metric calculates the summed absolute difference between the payments provided by the rule being examined and the reference. In the budget-balanced combinatorial-exchange examples we are about to consider, the budget constraint will render this summed difference identical across the rules for a fixed report (e.g. truth). This prevents L_1 from being useful for design, a point that we will show experimentally in Section 4.5. The L_∞ metric specifically targets the worst case within a given instance, and in a prior-free manner. It is thus the consistent standard *regret*-based condition that we discussed in Section 3.3.2.5. We therefore expect rules designed with this condition in mind, such as Threshold, to do well with respect to this metric.

4.3 Evaluation in Three CE Scenarios

We consider three CE instance generators, and thus three different problem scenarios. Two are variations on the combinatorial auction generators (*Decay* and *Uniform*) introduced in Sandholm [2002a]. To make these work in an exchange setting, we first fix the set of available goods and then distribute them to the selling agents, and the demand for them among the buying agents. With these endowments and 'demand sets' specified, we then choose negative seller (reserve) values, and positive buyer values for XOR bundles of items restricted to these endowments and 'demand sets', according to Sandholm's rules. The third is a new generator (*Super*), specifically designed for CEs, and with features carefully crafted for super-additive valuations. Here every good $g \in G$ is assigned a uniform random common value $c(g) \geq 0$, and a uniform random private value specific to agent i, $y_i(g) \geq 0$. Agent i then has a value for an individual good $w_i(g) = \beta y_i(g) + (1-\beta)c(g)$, for some β (.5 in our experiments). The value to agent i for all bundles of items $S \subseteq G_i$ is then $(\sum_{g \in S} w_i(g))^\gamma$, for some

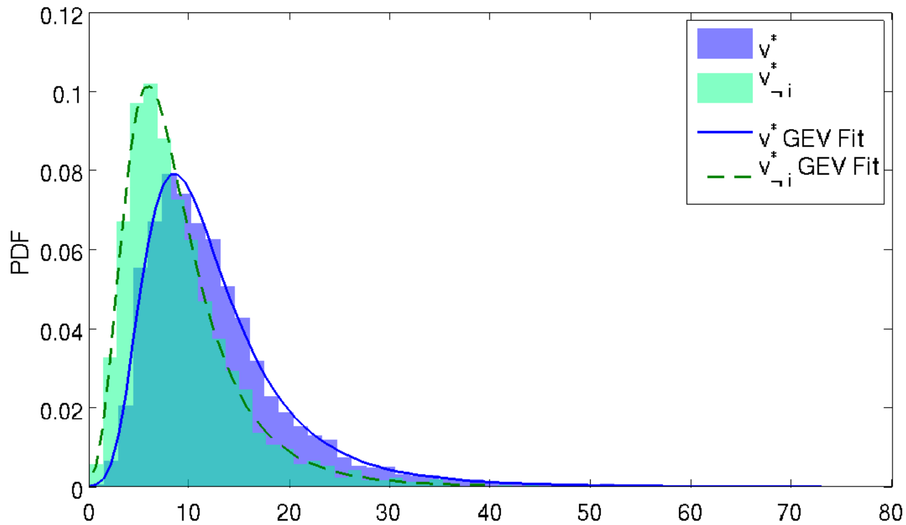


Figure 4.2: Distribution of surplus in the main economy and in the winning marginal economies using the Super generator; Also, an MLE fit of the GEV family to these distributions

$\gamma > 1$, where G_i is the endowment/'demand set' for agent i.[4] As above, this value forms a negative (reserve) value for sellers and a positive value for buyers.[5]

It is instructive to consider the distribution of $V^*(\boldsymbol{v})$, $V^*(\boldsymbol{v}_{\text{-}i})$, and the VCG payoff $V^*(\boldsymbol{v}) - V^*(\boldsymbol{v}_{\text{-}i})$ for trading agents that is induced by these generators. See Figures 4.2 and 4.3 for the *Super* distribution (the others are qualitatively similar). We can precisely identify the form of these distributions. Fix instance $\boldsymbol{v}$. Consider the set Λ of feasible trades in a given market instance. Each $\lambda \in \Lambda$ has a corresponding total value $V(\lambda, \boldsymbol{v})$, and $V^*(\boldsymbol{v})$ is by definition the maximum over these. Thus the V^* distribution is that of the extreme values of the underlying distribution of V. Such

[4]Note that we clear our markets with free disposal, with an implicit value for a bundle of items $S' \supseteq S$ equal to the value of S, unless explicitly specified as being worth more.

[5]We do not use CATS [Leyton-Brown and Shoham, 2006] for the generation of our data sets because its algorithms are explicitly designed for auctions and it is not straightforward to extend its distributions in a way that appropriately balances buyers and sellers. In the absence of such reference distributions, we have opted for these simpler existing generators, coupled with our own new generator.

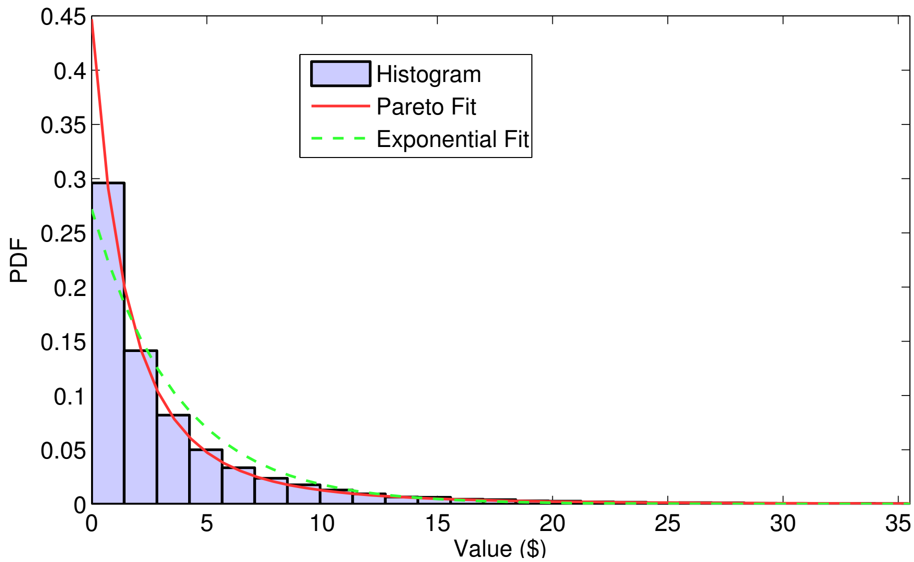


Figure 4.3: Distribution of VCG payoffs using the Super generator; Also, an MLE fit of the Exponential and Pareto families to this distribution

extreme value distributions have been extensively studied in the statistics literature, and can be precisely modeled by the *Generalized Extreme Value Distribution* (GEV) PDF

$$\frac{1}{\sigma}(1+\xi\frac{(x-\mu)}{\sigma})^{-1/\xi-1}e^{-(1+\frac{\xi(x-\mu)}{\sigma})^{-1/\xi}}, \tag{4.8}$$

parametrized by μ (location), σ (scale), and ξ (shape) [Coles, 2001]. Figure 4.2 shows the excellent fit of the GEV that can be produced for both V^* and V^*_{-i} via maximum likelihood estimation (MLE). The VCG payoff distribution is the distribution of exceedences (by V^*) over V^*_{-i}, and is well-modeled by a *Generalized Pareto Distribution* (GPD), though this model is typically motivated in cases of exceedences over a fixed threshold. The GPD has a PDF

$$\frac{1}{\sigma}(1+\frac{\xi(x-\mu)}{\sigma})^{-(1/\xi+1)}, \tag{4.9}$$

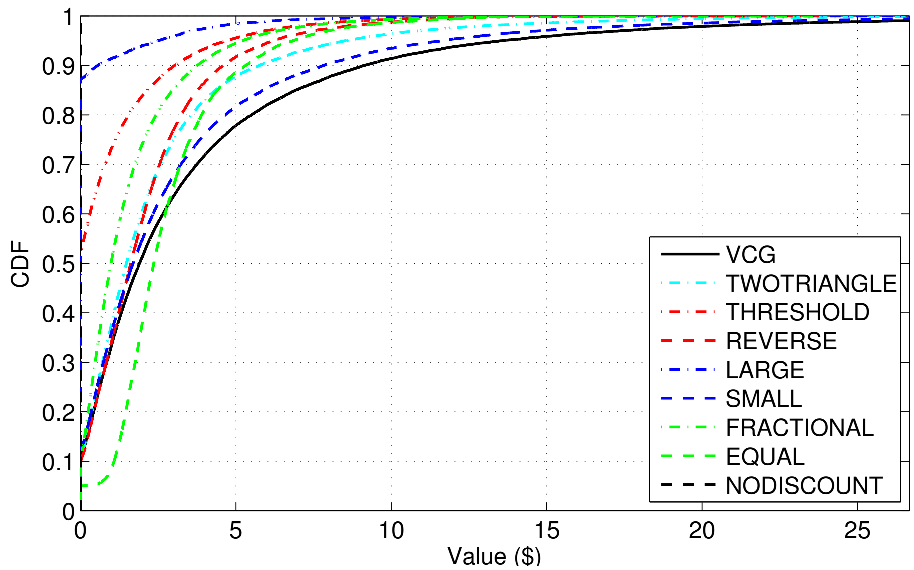


Figure 4.4: Distribution of agent payoffs in each mechanism

using a similar parametrization.[6] The MLE fit of the GPD is illustrated in Figure 4.3, along with the fit of a simple Exponential distribution (which is generalized by the GPD), indicating that the extra parameters of the GPD are improving the fit.

We can immediately consider how well each of the mechanisms performs at mimicking this distribution of payoffs. Figure 4.4 shows an empirical CDF of the payoff to trading agents under each of the mechanisms, when agents behave truthfully (again for the *Super* generator, the others being similar). One can visually confirm that the Small rule is the one best tracking the VCG payoffs in distribution. Table 4.1 evaluates the normalized metrics on each mechanism, computed over all three scenarios. Consistent with Figure 4.4, we can observe that Small has the smallest $KLnorm$ metric. On the other hand, Threshold has the smallest $L_2 norm$ and $L_\infty norm$ (regret-based) metrics. Notice that the $L_1 norm$ metric is identical across all rules except No Dis-

[6]Taking the cross-correlation between the distributions of V^* and V^*_{-i} will give the distribution of the difference under an assumption of independence, but independence will not hold here.

Mechanism	*KLnorm*	L_1 *norm*	L_2 *norm*	L_∞ *norm*
Two Triangle	0.0735	**0.5914**	0.3170	0.1917
Threshold	0.0472	**0.5914**	**0.2355**	**0.1016**
Reverse	0.1251	**0.5914**	0.3066	0.2210
Small	**0.0452**	**0.5914**	0.4208	0.3527
Large	0.0559	**0.5914**	0.3110	0.2070
Fractional	0.0741	**0.5914**	0.2528	0.1513
Equal	0.3043	0.8037	0.3727	0.2576
No Discount	0.6372	1.5876	0.6679	0.4030

Table 4.1: Metric value at truth averaged across all three CE scenarios. Minimal metric values in **bold**

count and Equal. This is because the other mechanisms always allocate all available surplus as payoff to agents.[7]

4.4 Equilibrium Analysis

Computing the equilibrium of the various mechanisms presents a challenge because this is an infinite game of incomplete information, with a continuum of possible valuations and thus possible agent strategies. The game also has combinatorial structure. There are at present no tractable methods to compute the exact Bayes-Nash equilibrium for such problems. The state of the art approach is to search for

[7]The L_1 *norm* metric differs for Equal only because it sometimes allocates an agent more payoff than the VCG payoff.

parametrized strategy profiles that constitute a restricted equilibrium through iterated best-response dynamics [Vorobeychik and Wellman, 2008].[8] We adopt a variation on this, where we place a heuristically-guided and annealing best-response search within a sampling procedure for determining the joint-strategy profile.

4.4.1 Computing Restricted Bayes-Nash Equilibrium

One simple restriction that one could impose is that every agent shaves its valuation by $\alpha \geq 0$, and thus seek a symmetric Bayes-Nash equilibrium. In the context of a CE, agents would report valuations $(1-\alpha)v$ and $(1+\alpha)v$ for buyers and sellers respectively (note that sellers have negative values.) This simplification realizes a one-dimensional, continuous strategy space.

We compute a more fine-grained equilibrium by also running experiments in which we adopt two or three shave factors. With multiple shave factors, we associate each agent in an instance endogenously with a valuation class depending on its valuation function. For example, with three shave factors α_1, α_2, and α_3, we sort agent valuations into "low," "medium" and "high" valuation classes, with an agent in each class associated with shave factor α_1, α_2 and α_3 respectively. We then search for an equilibrium defined in terms of these three parameters. To sort agent valuations, we first draw a number of samples of otherwise unused agents from the same distribution that defines the CE scenario, and for each of these agents, we record the 95^{th} percentile of value across the trades that define its valuation function. An agent's valuation class is identified by comparing the value at the 95^{th} percentile on the trades in its valuation

[8] An exact solver exists only for two-player games with one-dimensional private valuations, based on a piecewise-linear strategy representation [Reeves and Wellman, 2004].

with the sampled values, and assigning a class according to placement in the lower, middle, or upper third (tritile) of this sampled distribution. This self classification process corresponds to an information assumption: agents have a good idea about which tritile of the value distribution to which they belong.

Our algorithm for finding the equilibrium begins with provisional shave factors $\{\widehat{\alpha}_k\}$ (e.g., for $k \in \{1, 2, 3\}$) set to 0. It then repeatedly generates a set of CE instances from the particular distribution (Uniform, Decay or Super), and for each instance, each agent is placed into a valuation class. In each iteration t of the algorithm, and for each agent i, a grid search is performed on α-values to find its best-response value $\widetilde{\alpha}_i$, while using provisional α-values assigned to the other agents. For each valuation class, the provisional $\widehat{\alpha}_k$ are then updated as $\widehat{\alpha}_k^{t+1} := \theta\widehat{\alpha}_k^t + (1-\theta)\overline{\alpha}_k^t$, where $\theta = .5$ and $\overline{\alpha}_k^t$ is the mean of the best response values $(\widetilde{\alpha}_i)$ in iteration t calculated for each agent associated with the class k. The width of the grid search in period $t+1$ is chosen endogenously, with 10 points covering a span of $|\widehat{\alpha}_k^t - \overline{\alpha}_k^t|$. This span is also used as an error estimation and search stops when it falls below a fixed constant, 0.001 in the experiments.

4.4.2 Equilibrium: Results

Table 4.2 shows the results with one-dimensional and three-dimensional strategy spaces (respectively "one class" and "three classes"), for all three generators. In the case of three classes, the reported shave factor is the average across $\{\alpha_1, \alpha_2, \alpha_3\}$. The best mechanisms in each case are indicated in **bold**. Surprisingly, the Threshold mechanism, which has theoretical support in minimizing the *ex post* regret across all

	One Equilibrium Class						Three Equilibrium Classes					
	Shave Factor			**Efficiency** (%)			**Shave Factor**			**Efficiency** (%)		
Rule	**Dec.**	**Uni.**	**Sup.**	**Dec.**	**Uni.**	**Sup.**	**Dec.**	**Uni.**	**Sup.**	**Dec.**	**Uni.**	**Sup.**
VCG	*0.0*	*0.0*	*0.0*	*100*	*100*	*100*	*0.0*	*0.0*	*0.0*	*100*	*100*	*100*
Two Triangle	**0.1**	**0.2**	**0.6**	**99.99**	**100**	**99.99**	**0.1**	**0.4**	**5.6**	**99.99**	**100**	**97.95**
Threshold	12.0	28.7	10.7	99.09	97.43	98.01	14.6	27.2	11.2	93.64	81.09	89.74
Reverse	14.9	57.7	52.3	98.70	83.38	51.52	13.0	65.8	57.6	98.99	77.30	56.08
Small	**0.1**	**0.2**	**0.3**	**99.99**	**100**	**100**	**0.0**	**0.1**	**0.2**	**99.99**	**100**	**100**
Large	**2.6**	**2.3**	9.8	**99.96**	**99.99**	98.26	**2.8**	**2.9**	67.1	**99.96**	**99.98**	78.83
Fractional	71.2	71.1	53.0	59.39	67.34	49.07	62.7	81.9	62.0	37.12	63.09	56.77
Equal	75.4	77.6	52.5	51.96	55.76	51.01	62.2	78.3	66.8	33.35	54.21	52.19
No Discount	75.6	76.0	53.2	51.56	59.01	48.23	62.3	80.9	72.4	34.15	50.11	48.21

Table 4.2: Restricted Bayes-Nash equilibrium: Shave Factor and Allocative Efficiency in Each Mechanism

these mechanisms, does not perform nearly as well as the Small mechanism, either in terms of the size of shave factor (close to zero indicates approximate incentive-compatibility) or the resulting allocative efficiency. Recall that the Small mechanism is also the one with the lowest KL-divergence metric.[9]

[9] One interesting anomaly in the data is for Large between the "one class" and "three class" analysis. With one class, a balance must be made in the equilibrium between those agents with high valuations (likely to receive their full discount without any shave under Large) vs. those with low valuations (unlikely to receive any discount without shaving). In this case, the former constrains the latter and agents choose not to shave much in equilibrium. But with three shave factors there is increased discrimination, and the optimal shave for those with small valuations becomes very extreme. This, coupled with the fact that there are large numbers of small discounts relative to a few large discounts, decreases the efficiency of the Large rule in equilibrium.

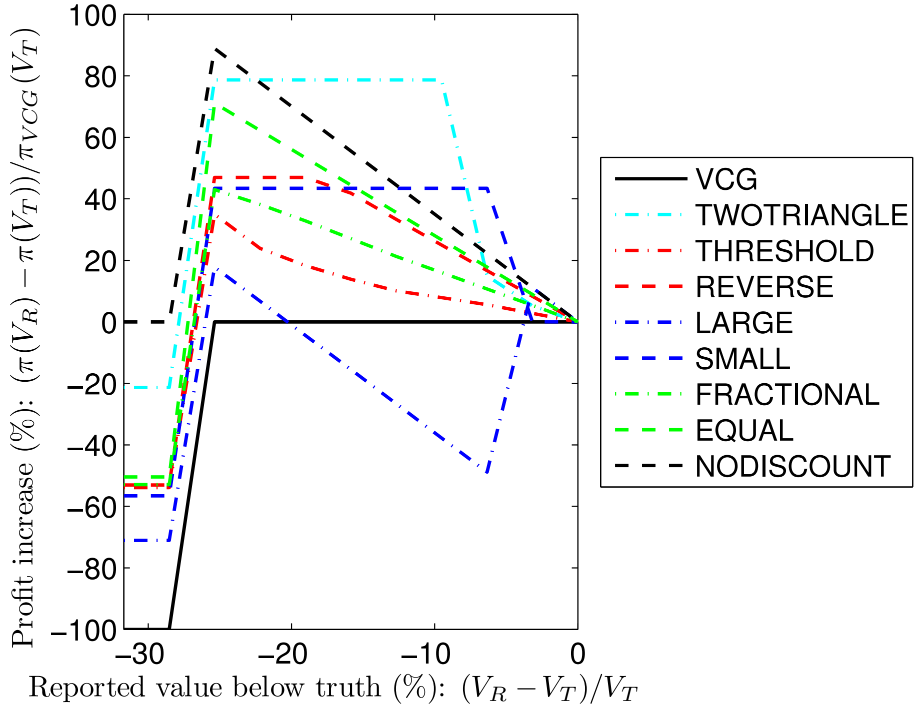


Figure 4.5: Profit gain by unilateral misreport

4.4.2.1 Unilateral Deviation

To understand the effect of the Small payment rule, which allocates payment preferentially to agents with a small VCG payoff, we can study an individual agent's incentive to deviate. Figure 4.5 shows the profit gained by a single agent in a representative single instance drawn from the *Super* scenario, as the agent reports V_R compared to truth V_T for its winning trade and 0 for all other trades, under each of the mechanisms. The profit is normalized to its maximal possible profit, i.e. its VCG profit, and the experiment considers only unilateral deviation by this agent with all other agents reporting truthfully. The agent in question has a large payoff under VCG, which the Large mechanism fully allocates. As the agent deviates he suffers a loss under the Large mechanism. Under all the other mechanisms (except VCG)

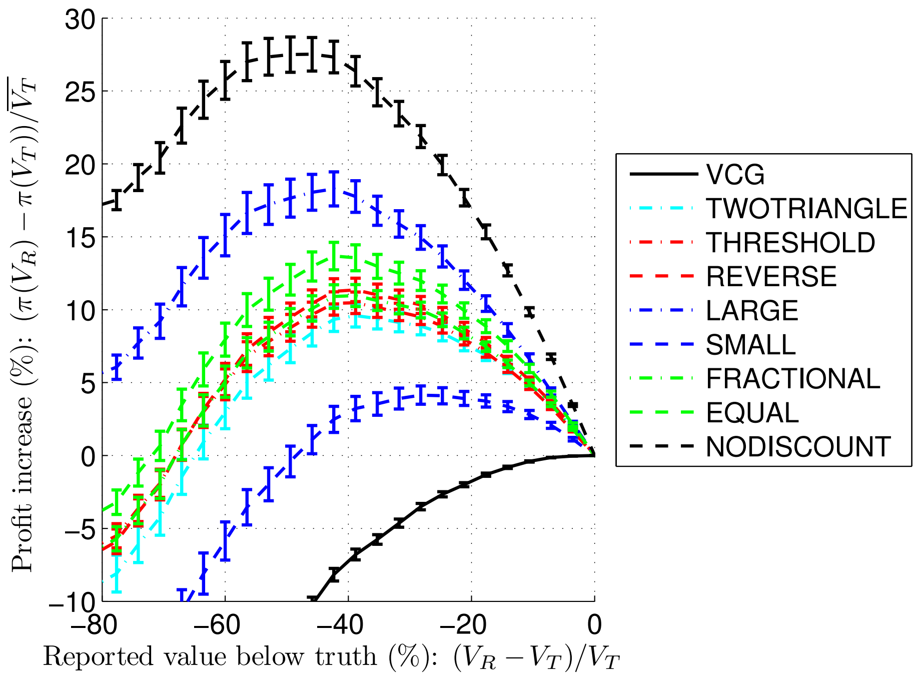


Figure 4.6: Expected profit by unilateral mis-report

there is at least some gain from deviation. Unlike the other rules, though, the Small mechanism exhibits a flat plateau once the agent deviates by a small amount. Thus the incentives to deviate significantly can be quite low under Small, even for agents whose payoff in VCG is quite large.

This analysis represents only a single agent in a single instance. In order to get a more comprehensive picture we can average several thousand such single-instance trajectories, as shown in Figure 4.6. Here we see that mis-reporting makes an agent strictly worse-off under VCG, as expected. But importantly, we see that the Small mechanism provides only a small expected gain from deviation, and the maximal expected gain occurs with less shaving than the other mechanisms. While still a non-equilibrium analysis (other agents are truthful), this is suggestive of the good equilibrium performance under Small.

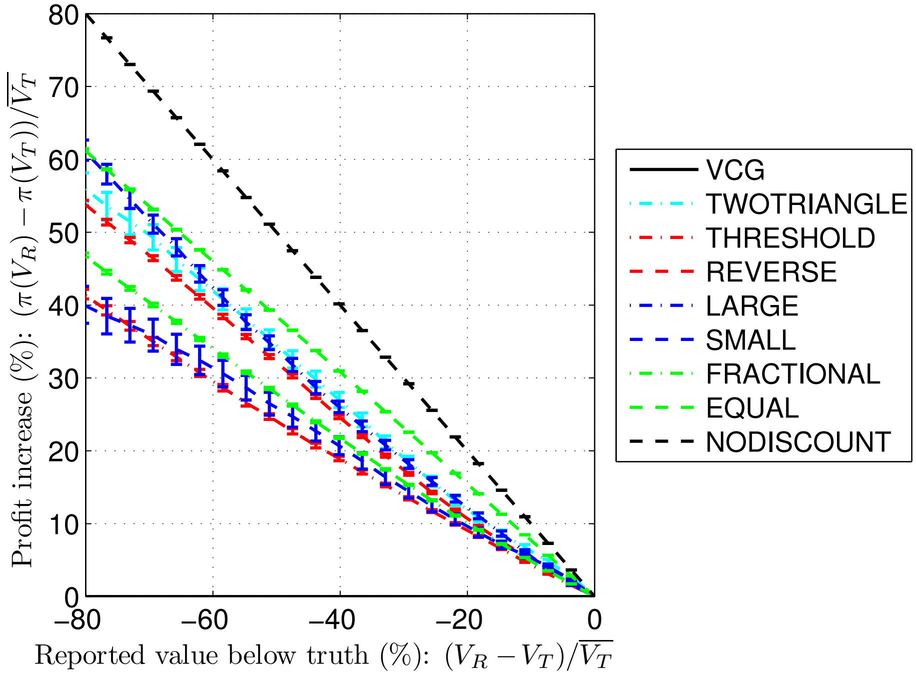


Figure 4.7: Conditional profit by unilateral mis-report

In determining a good strategy, an agent is in essence making an *ex ante* trade-off between potential gain from a successful manipulation and potential loss given an unsuccessful manipulation. By further conditioning on those mis-reports that are successful (i.e., when an agent still trades) and unsuccessful, we arrive at Figures 4.7 and 4.8. We see that Small is near the bottom of the pack for both conditional gain and conditional loss, indicating that success brings relatively less gain while failure brings relatively more pain than in other mechanisms. In comparison, an unsuccessful manipulation does not hurt an agent as much under the Threshold mechanism, contributing to its weaker equilibrium performance.

Remark. Unlike Small, the Threshold mechanism tends to allocate payoff to fewer agents, and with very few (if any) agents receiving their maximal payoff. This is driving the divergence from the VCG payoff distribution and also this larger loss

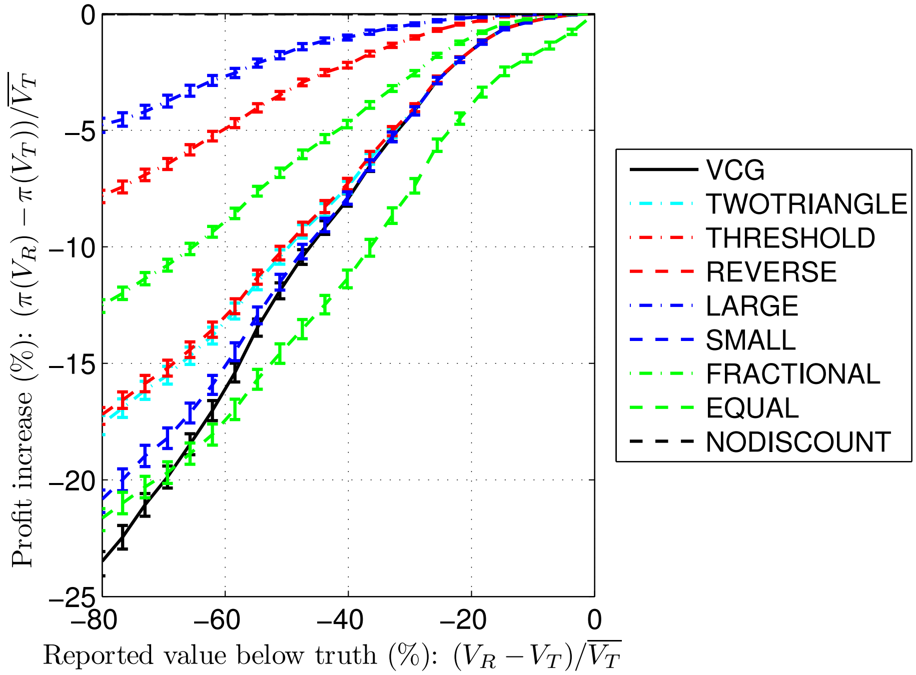


Figure 4.8: Conditional loss by unilateral mis-report

in payoff, conditioned on an unsuccessful manipulation. By making the distribution on payoffs close to the reference, VCG mechanism, the Small mechanism makes the expected payoff, conditioned on success and failure, both relatively close to the profile under VCG (compared to the other mechanism rules); i.e., close to zero for success and close to forfeiting the maximal payoff for failure. Since the VCG payoff distribution is skewed such that many agents have only small opportunities for gain (see Figure 4.3), then many of these opportunities can be addressed by the Small mechanism with the remaining opportunities for gain entailing significant risk.

However, Small has some rather perverse properties as well. In particular, those agents that under VCG would be entitled to the largest discounts receive no discount at all. This means that these agents will be asked to pay significantly more than their marginal contribution to the outcome, as VCG would charge. This interpretation of

VCG is particularly appealing, as a payment of your marginal impact would seem in some profound sense highly appropriate – and Small violates this significantly. However, perhaps even more importantly, Small offers high discount players zero profit as it charges them exactly their bid (i.e. gives them no discount) and, if they have reported truthfully, their discount will be the profit on their trade. This means that when agents bid truthfully it will offer those agents that ostensibly should be making the most profit, zero profit, and those who should be making the least profit their full measure. While we have shown this to result in good incentive properties, this may come as little solace to a trader forced to break even on a trade that under other rules would be hugely profitable.

4.5 Metric Analysis

In this section we adopt the correlation between each metric and the equilibrium shave factor and efficiency as a measure of the informativeness of the metric in quantifying the degree of strategyproofness of a mechanism. The correlation is determined over a data set of several thousand instances. For each generator (Uniform, Decay and Super) there are 6 mechanisms[10] and 3 different equilibrium analyses (for 1, 2 and 3 shave factors.) This provides 3 x 6 x 3 = 54 data points, with the average efficiency, average shave factor, and metric computed for each and enabling a correlation to be computed. The results are presented in Table 4.3. We only present results for normalized metrics throughout this section because they dominate in terms of

[10] We drop Equal, No Discount and VCG from this correlation analysis; No Discount and VCG are not in the candidate class of mechanisms, and Equal is outside the class we are especially interested in because it sometimes allocates an agent more than its VCG payoff.

Correlation with Efficiency at Truth			
Metric	**Corr.**	**ρ-value**	**Significant?**
KLnorm	-0.3814	0.0044	Y
L_1 *norm*	-0.1698	0.2197	N
L_2 *norm*	0.0154	0.9120	N
L_∞ *norm*	0.0220	0.8745	N
Correlation with Mean Shave at Truth			
Metric	**Corr.**	**ρ-value**	**Significant?**
KLnorm	0.3794	0.0047	Y
L_1 *norm*	0.1610	0.2447	N
L_2 *norm*	-0.1001	0.4712	N
L_∞ *norm*	-0.1147	0.4087	N

Table 4.3: Correlation between metrics evaluated at truth and both efficiency and the amount of shaving, considering all 54 conditions (Significance at 0.05 level)

statistical significance. *We see that the KL-norm metric is negatively correlated with efficiency and positively correlated with the equilibrium shave factor. In both cases this correlation is significant at the 0.05 level, whereas the correlation for the other, regret-based metrics is not significant.* This finding is consistent with the theoretical reasoning that will be developed in Section 4.7.2.

Although of secondary importance, we can also consider the informativeness of each metric in *validating* how close to truthful an equilibrium is, based only on observed data in the equilibrium. This is interesting, for example, in evaluating the

Correlation with Efficiency in Equilibrium			
Metric	**Corr.**	**ρ-value**	**Significant?**
KLnorm	-0.4989	1.2292e-04	Y
L_1*norm*	-0.6460	1.3269e-07	Y
L_2*norm*	-0.5119	7.6150e-05	Y
L_∞*norm*	-0.3762	0.0051	Y
Correlation with Mean Shave in Equilibrium			
Metric	**Corr.**	**ρ-value**	**Significant?**
KLnorm	0.2702	0.0482	Y
L_1*norm*	0.5870	3.0820e-06	Y
L_2*norm*	0.4615	4.4464e-04	Y
L_∞*norm*	0.3738	0.0054	Y

Table 4.4: Correlation between metrics evaluated at equilibrium and both the efficiency and the amount of shaving, considering all 54 conditions (Significance at 0.05 level)

degree of strategyproofness of a mechanism based only on observed, equilibrium behavior. The correlation data, evaluated over the same 54 conditions but now in equilibrium for each mechanism, is presented in Table 4.4. We find that the L_1*norm* is more informative, in equilibrium, than the *KLnorm* and other metrics. A strong, and significant correlation is also found for the L_2*norm* metric. The L_1*norm* measures the average (normalized) regret of an agent. Our hypothesis for why the average equilibrium regret is effective in this regard, is that the further a mechanism is from being strategyproof, the further agents will deviate from truthful bidding in equilib-

Mechanism	*KLnorm*	L_1*norm*	L_2*norm*	L_∞*norm*
Two Triangle	0.0820	0.6096	0.3271	0.1976
Threshold	0.0556	0.6991	**0.2984**	**0.1367**
Reverse	0.1421	0.9415	0.4896	0.3104
Small	**0.0452**	**0.5903**	0.4208	0.3534
Large	0.0668	0.8269	0.4494	0.2916
Fractional	0.1303	1.1456	0.5683	0.3477
Equal	0.2033	1.3758	0.7291	0.4919
No Discount	0.3114	1.9962	1.0311	0.6721

Table 4.5: Metric value at equilibrium averaged across all three scenarios and valuation classes. Minimal values are in **bold**

rium, and the more mistakes (*ex post*) will occur. *Note, though, that the L_1norm metric does not provide guidance for design because it requires a designer to reason about properties in equilibrium. In fact, for a fixed distribution on agent reports (e.g., at truth) almost all of the mechanisms have the* same *L_1norm metrics (see Table 4.1).*

In Table 4.5 we present the various metrics evaluated at the equilibrium of each mechanism over the 54 conditions. Here, it is apparent that Small is most effective at minimizing L_1*norm*, i.e., in minimizing the average regret faced by agents in equilibrium.

In contrast, and counter to accepted wisdom, the Threshold rule (which is designed to minimize maximal regret given reports) has higher average regret in equilibrium. The Threshold rule is most effective in minimizing the L_2*norm* and L_∞*norm* metrics, which is perhaps unsurprising given its design.

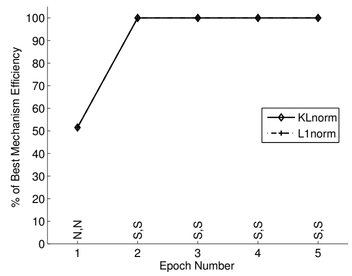
4.6 Online Mechanism Selection

In this section, we adopt a straw-man experiment to understand the effectiveness of the various metrics in guiding an online search for the best mechanism, using only information that is available to an observer in equilibrium play. Note that a simpler question about heuristic design was already answered earlier: the Small mechanism has the best *KLnorm* metric, and thus would be adopted as the best mechanism design under this lens. But here we ask a different question: given observed equilibrium play, is the *KLnorm* metric effective in suggesting a new mechanism to switch to? The set-up is one of online search. We do not get to evaluate the counter-factual equilibrium that would exist under each candidate equilibrium, nor the true, underlying efficiency of an equilibrium. The only data that is available is based on observing the equilibrium bids, allocations and payments in a current mechanism.

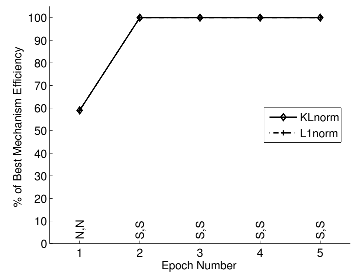
The online search is instantiated for a particular metric and proceeds as follows. The search takes place over a sequence of epochs, with a single mechanism deployed in each epoch and an epoch consisting of a fixed number of CE instances. The search is initialized somehow (here we always initialize to the No Discount mechanism.) An epoch provides two kinds of data. For the mechanism that is used, it provides distributional information about the equilibrium, bids and the metric can be evaluated on the (revealed) payoffs received by agents. But it is also possible to take the same distribution on bids, and evaluate the metric for each of the other available mechanisms. That is, take the bids as fixed and simply evaluate the metric on the payoffs that would be induced by the other mechanisms (and ignoring that the input is actually the equilibrium for the current mechanism, and not the truthful distribution).

At the end of each epoch, we evaluate each metric based on the data collected in the equilibrium of the current mechanism and switch to the mechanism with the lowest metric. In evaluating the metrics, we retain data from previous runs of the same mechanism as adopted in the current epoch, enabling ever more accurate metrics to be calculated. The only caveat is that we check for cycles and break them as follows: e.g., suppose we are presently using mechanism A and the metric over the data under A indicates mechanism B to be best, but B has been selected in the past and the data under mechanism B indicates that mechanism A is best. If such a cycle is found, then the online search proceeds by evaluating the metric on A and B over the *combined* data set from running both A and B in the past and selecting the best. We expect the data obtained from within the play of a mechanism itself to be the best indicator of its performance, and thus we use this combined data set only for breaking such cycles.

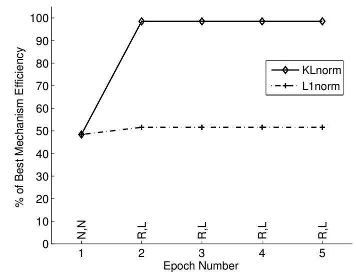
Figure 4.9 shows the results of running this algorithm for each of the three different CE scenarios and for both 1 and 3 agent classes in defining the simulated equilibrium. We compare the performance of the algorithm with the *KLnorm* and L_1 *norm* (average regret) metrics. Each graph shows the epoch on the x-axis and the efficiency of the chosen rule as a fraction of the ideal rule (Small) on the y-axis; the epoch size was set to 100 for these experiments. Online search with the *KLnorm* metric very quickly chooses a good rule, and with performance that tends to dominate that of search with the L_1 *norm* metric. Performance of the L_2 *norm* and L_∞ *norm* is nearly identical to that of L_1 *norm*, and is thus omitted for clarity. The online search performs least well in the Super scenario 3 class case, where it chooses to leave the Small rule for Large



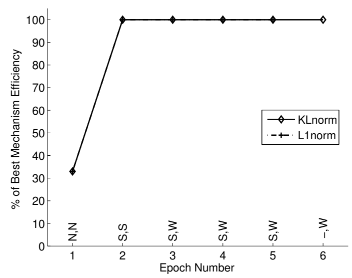
(a) Decay, 1 Class



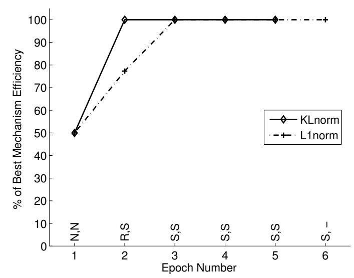
(b) Uniform, 1 Class



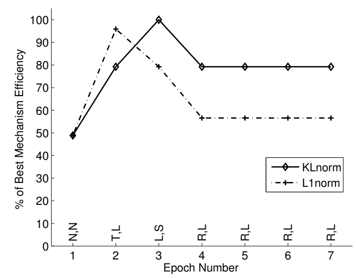
(c) Super, 1 Class



(d) Decay, 3 Class



(e) Uniform, 3 Class



(f) Super, 3 Class

Figure 4.9: Online selection: choosing the mechanism algorithmically. The labels along the x-axis indicate the rule chosen in a given epoch under the *KLnorm* and L_1 *norm* rules respectively

based on the data available after epoch 3 and then fails to return. From within its own equilibrium the Large rule looks promising and the ideal Small rule is extremely different in effect and distribution– making escaping the Large local-maxima difficult.

4.7 Metrics as Bounds

We now turn to integrating the metric-based approach covered so far in this chapter with the design criteria we proposed in Chapter 3. In this section, we offer a proof that the KL-Divergence formally bounds one of these target criteria.

Suppose we are choosing an approximately strategyproof mechanism, and let's assume we, as the designer, have access to a prior over the joint distribution of agent values (from an analytic model, or backed out from historical data). As a designer, we seek to either minimize the sum of expected gain from equations (3.7) or (3.13), or the sum of the worst case deviations from equations (3.8) or (3.14) As we have a full distribution, as a 'risk-neutral' designer we will adopt as a goal good performance in expectation, and therefore we choose the expected case over the worst-case approximate incentive-compatibility criterion. And thus we take the expected case *ex ante* as the target in what follows, as we believe it to be a very good proxy for full BNE behavior.

4.7.1 Bounding Unilateral Gain with a Reference Mechanism

Suppose we have an individually rational mechanism $m = (\mathcal{W}, p)$, with an outcome rule $\mathcal{W}$ and a payment rule $p(\boldsymbol{v}) \in \mathbb{R}^n$, that we know is *not* strategyproof for agent i. Further, suppose we also have a strategyproof reference mechanism $m^* = (\mathcal{W}, p^*)$ with the same outcome rule but a different payment rule. This mechanism is optimal with respect to the incentive property, but fails some other other property of m that we find desirable (e.g. budget balance). Further, we assume that m is *reference-bounded* by m^*, meaning that $p_i(\boldsymbol{v}) \geq p_i^*(\boldsymbol{v}) \; \forall \; \boldsymbol{v}$. We note that all of the rules defined by Parkes et al. [2001a] (see Section 3.2.1) obey this property in the CE environment when using VCG as the reference. While there are clearly rules outside of this class, this restriction does not appear particularly onerous. Intuitively, one obvious way for mechanisms to achieve incentive compatibility is to 'pay' agents to be truthful by careful construction of the agent discounts, which is one way to interpret what the

VCG payment rule does. With these definitions we can state the following theorem:

Theorem 4.1: *Given a strategyproof reference mechanism m^* and an individually rational reference-bounded mechanism m with an identical outcome rule that is non-strategyproof for agent i, the expected ex ante gain in profit from unilateral deviation for agent i, in the sense of equation* (3.7)*, is bounded from above by agent i's expected excess profit in m^* over m. Formally:*

$$\epsilon_{EA,i} \equiv \mathop{\mathbb{E}}_{v_i}\left[\max_{v_i'}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_i(\boldsymbol{v})\right]\right]\right] \leq \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[\pi_i^*(\boldsymbol{v})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[\pi_i(\boldsymbol{v})\right] \quad (4.10)$$

Proof. First we expand:

$$\mathop{\mathbb{E}}_{v_i}\left[\max_{v_i'}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[v_i(\mathcal{W}(v_i'; \boldsymbol{v}_{\text{-}i})) - p_i(v_i'; \boldsymbol{v}_{\text{-}i})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[v_i(\mathcal{W}(\boldsymbol{v})) - p_i(\boldsymbol{v})\right]\right]\right]$$
$$\leq \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[v_i(\mathcal{W}(\boldsymbol{v})) - p_i^*(\boldsymbol{v})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[v_i(\mathcal{W}(\boldsymbol{v})) - p_i(\boldsymbol{v})\right] \quad (4.11)$$

Now we rearrange the expectations:

$$\mathop{\mathbb{E}}_{v_i}\left[\max_{v_i'}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[v_i(\mathcal{W}(v_i'; \boldsymbol{v}_{\text{-}i})) - p_i(v_i'; \boldsymbol{v}_{\text{-}i})\right]\right]\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[v_i(\mathcal{W}(\boldsymbol{v}))\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[p_i(\boldsymbol{v})\right]$$
$$\leq \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[p_i(\boldsymbol{v})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[p_i^*(\boldsymbol{v})\right] \quad (4.12)$$

Or:

$$\mathop{\mathbb{E}}_{v_i}\left[\max_{v_i'}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[v_i(\mathcal{W}(v_i'; \boldsymbol{v}_{\text{-}i})) - p_i(v_i'; \boldsymbol{v}_{\text{-}i})\right]\right]\right]$$
$$+ \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[p_i^*(\boldsymbol{v})\right] - 2\mathop{\mathbb{E}}_{\boldsymbol{v}}\left[v_i(\mathcal{W}(\boldsymbol{v}))\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[p_i(\boldsymbol{v})\right] \leq 0 \quad (4.13)$$

Because equality can always be achieved by setting $v_i' = v_i$, we have:

$$\mathop{\mathbb{E}}_{v_i}\left[\max_{v_i'}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[v_i(\mathcal{W}(v_i'; \boldsymbol{v}_{\text{-}i})) - p_i(v_i'; \boldsymbol{v}_{\text{-}i})\right]\right]\right] \geq \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[v_i(\mathcal{W}(\boldsymbol{v})) - p_i(\boldsymbol{v})\right] \quad (4.14)$$

Now add (4.13) and (4.14):

$$\mathop{\mathbb{E}}_{\boldsymbol{v}}[p_i^*(\boldsymbol{v})] - 2\mathop{\mathbb{E}}_{\boldsymbol{v}}[v_i(\mathcal{W}(\boldsymbol{v}))] - \mathop{\mathbb{E}}_{\boldsymbol{v}}[p_i(\boldsymbol{v})] + \mathop{\mathbb{E}}_{\boldsymbol{v}}[v_i(\mathcal{W}(\boldsymbol{v}))] - \mathop{\mathbb{E}}_{\boldsymbol{v}}[p_i(\boldsymbol{v})] \leq 0 \tag{4.15}$$

And simplify:

$$\mathop{\mathbb{E}}_{\boldsymbol{v}}[p_i^*(\boldsymbol{v})] - \mathop{\mathbb{E}}_{\boldsymbol{v}}[v_i(\mathcal{W}(\boldsymbol{v}))] - 2\mathop{\mathbb{E}}_{\boldsymbol{v}}[p_i(\boldsymbol{v})] \leq 0 \tag{4.16}$$

Or:

$$\mathop{\mathbb{E}}_{\boldsymbol{v}}[p_i^*(\boldsymbol{v}) - p_i(\boldsymbol{v})] \leq \mathop{\mathbb{E}}_{\boldsymbol{v}}[\pi_i(\boldsymbol{v}))] \tag{4.17}$$

Now because m is *reference-bound* by m^* the left side must be weakly less than zero. The right side is the expected profit under m, which must be weakly greater than zero as m is individually rational.[11] □

This theorem indicates that simply taking the difference in expected profit between a reference mechanism and a mechanism under evaluation should be informative of the *ex ante* unilateral gains available in the mechanism.

4.7.2 Bounding Excess Profit Between Mechanisms with a Metric

Suppose we have two mechanisms $m_A = (\mathcal{W}_A, p_A)$ and $m_B = (\mathcal{W}_B, p_B)$, with possibly different outcome functions and payment rules. Now, consider the quantity:

$$\epsilon_{A:B,i} \equiv \mathop{\mathbb{E}}_{\boldsymbol{v}}[\pi_{A,i}(\boldsymbol{v}) - \pi_{B,i}(\boldsymbol{v})] \tag{4.18}$$

which is the expected difference in profit for agent i between the two mechanisms while being truthful.

[11] As a corollary of Theorem 4.1, we also get that $\mathbb{E}_{v_i}\left[\max_{v_i'}\left[\mathbb{E}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i})\right]\right]\right] \leq \mathbb{E}_{\boldsymbol{v}}\left[\pi_i^*(\boldsymbol{v})\right]$ under the same conditions by subtracting $\mathbb{E}_{\boldsymbol{v}}\left[\pi_i(\boldsymbol{v})\right]$ from both sides of (4.10).

We can define versions of the profit functions that are functions solely of agent i's report by embedding the expectations:

$$\breve{\pi}_{A,i}(v_i) = \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_{A,i}(\boldsymbol{v})\right] \tag{4.19}$$

$$\breve{\pi}_{B,i}(v_i) = \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_{A,i}(\boldsymbol{v})\right] \tag{4.20}$$

Next, by using the PDF of the value distribution for agent i, f_{v_i}, we can find the PDFs of $\breve{\pi}_{A,i}$ and $\breve{\pi}_{B,i}$, using the same change-of-variables technique as in (3.10):

$$f_{\breve{\pi}_{A,i}}(x) = \left|\frac{1}{\breve{\pi}'_{A,i}(\breve{\pi}^{\text{-}1}_{A,i}(x))}\right| \cdot f_{v_i}(\breve{\pi}^{\text{-}1}_{A,i}(x)) \tag{4.21}$$

$$f_{\breve{\pi}_{B,i}}(x) = \left|\frac{1}{\breve{\pi}'_{B,i}(\breve{\pi}^{\text{-}1}_{B,i}(x))}\right| \cdot f_{v_i}(\breve{\pi}^{\text{-}1}_{B,i}(x)) \tag{4.22}$$

where the prime denotes the derivative operator and $z^{\text{-}1}$ denotes the inverse of function z. These are the probability distributions of the profit for being truthful when other agents bid according to $\boldsymbol{v}_{\text{-}i}$, under each mechanism. We can further define CDFs for these distributions as:

$$F_{\breve{\pi}_{A,i}}(x) = \int_0^x f_{\breve{\pi}_{A,i}}(x)dx \tag{4.23}$$

$$F_{\breve{\pi}_{B,i}}(x) = \int_0^x f_{\breve{\pi}_{B,i}}(x)dx \tag{4.24}$$

where we start the integration at 0 because we know that the profit distributions will only have positive support. Applying these definitions to equation (4.18) we obtain:

$$\epsilon_{A:B,i} = \int_0^\infty x f_{\breve{\pi}_{A,i}}(x)dx - \int_0^\infty x f_{\breve{\pi}_{A,i}}(x)dx \tag{4.25}$$

Next we take advantage of the following lemma:[12]

[12] For CDFs with full support, which we don't need here, we have: $\mathbb{E}(X) = \int_{-\infty}^{\infty} t f_X(t)dt = -\int_{-\infty}^{0} F_X(t)dt + \int_0^\infty 1 - F_X(t)dt$.

Lemma 4.2: *For all random variables X with non-negative support, PDF f_X and CDF F_X, we have:* $\mathbb{E}(X) = \int_0^\infty t f_X(t)dt = \int_0^\infty [1 - F_X(t)]\, dt$.

Proof. We start with the standard formula for integration by parts

$$\int_a^b fgdx = \left[f \int gdx\right]_a^b - \int_a^b \left(\int gdx\right) df$$

Now let $G_X(t) = 1 - F_X(t)$ so $G_X'(t) = -f_X(t)$ and let $H(t) = t$ so $H'(t) = 1$. Then we can substitute G_X for f and H' for g:

$$\int_a^b G_X(t)H'(t)dt = \left[G_X(t) \int H'(t)dt\right]_a^b - \int_a^b \left(\int H'(t)dt\right) G_x'(t)$$

or

$$\int_a^b G_X(t)H'(t)dt = \left[G_X(t)H(t)\right]_a^b - \int_a^b G_x'(t)H(t)$$

By substituting back for the original functions we obtain:

$$\int_a^b 1 - F_X(t)dt = \left[tG_X(t)\right]_a^b + \int_a^b tf_X(t)dt$$

Expanding the first term on the right side we obtain:

$$\left[tG_X(t)\right]_a^b = bG_X(b) - aG_X(a)$$

As F_X is a CDF we have:

$$\lim_{a\to 0} aG_X(a) = 0 \qquad \text{and} \qquad \lim_{b\to\infty} bG_X(b) = 0$$

So the term goes to zero when we are dealing with a probability distribution and we choose appropriate limits, obtaining:

$$\int_0^\infty [1 - F_X(t)]\, dt = \int_0^\infty tf_X(t)dt$$

□

Example 4.1: *Exponential Distribution*

$$
\begin{aligned}
f_X(t) &= \lambda e^{-\lambda t} & & & F_X(t) &= 1 - e^{-\lambda t} \\
\mathbb{E}[X] &= \int_0^\infty t f_X(t) dt & &\stackrel{?}{=} & \mathbb{E}[X] &= \int_0^\infty \left[1 - F_X(t)\right] dt \\
&= \int_0^\infty t(\lambda e^{-\lambda t}) dt & &\stackrel{?}{=} & &= \int_0^\infty \left[1 - \left(1 - e^{-\lambda t}\right)\right] dt \\
&= \frac{1}{\lambda}\int_0^\infty u e^{-u} du & &\stackrel{?}{=} & &= \int_0^\infty e^{-\lambda t} dt \\
&= \frac{1}{\lambda}(1) & &\stackrel{?}{=} & &= -\frac{1}{\lambda}(0-1) \\
&= \frac{1}{\lambda} & &= & &= \frac{1}{\lambda}
\end{aligned}
$$

□

Applying this lemma to equation (4.25), we obtain:

$$\epsilon_{A:B,i} = \int_0^\infty \left[1 - F_{\breve{\pi}_{A,i}}(x)\right] dx - \int_0^\infty \left[1 - F_{\breve{\pi}_{B,i}}(x)\right] dx \tag{4.26}$$

Or:

$$\epsilon_{A:B,i} = \int_0^\infty \left[F_{\breve{\pi}_{A,i}}(x) dx - F_{\breve{\pi}_{B,i}}(x)\right] dx \tag{4.27}$$

Note that the brackets here are important: the infinite integral of a CDF is infinite. We can insert an absolute value by relaxing to an inequality:

$$\epsilon_{A:B,i} \leq \int_0^\infty \left|F_{\breve{\pi}_{A,i}}(x) dx - F_{\breve{\pi}_{B,i}}(x)\right| dx \tag{4.28}$$

Which is the definition of the 1-Wasserstein metric $\mathcal{W}(\mu, \nu)$ (sometimes called the Kantorovich metric) for 1 dimensional distributions μ, ν [Villani, 2009], obtaining:

$$\epsilon_{A:B,i} \leq \mathcal{W}_1(f_{\breve{\pi}_{A,i}}, f_{\breve{\pi}_{A,i}}) \tag{4.29}$$

The Wasserstein metric is equivalent to the minimal transport problem, or 'earth-mover-distance'. If we envision the PDFs of each distribution as a pile of dirt, the Wasserstein metric is the amount of dirt that needs to be moved times the distance it must be moved to convert one pile into the other. For one-dimensional distributions it takes on this simple form involving only the CDFs M and N of μ and ν:

$$\mathcal{W}_1(\mu, \nu) = \int_{-\infty}^{\infty} |M(x) - N(x)| dx$$

For multi-dimensional distributions its calculation is far more involved, requiring us to find the minimal coupling between the distributions:

$$\mathcal{W}_k(\mu, \nu) = \sqrt[k]{\inf \mathbb{E}[|M - N|^k]}$$

where the infimum is over all joint distributions of M and N, where the marginal on M is μ and the marginal on N is ν.

Gibbs and Su [2002] relate the Wasserstein metric to the better known KL-divergence $\mathcal{K}(\mu, \nu)$ for probability functions on a bounded domain Ω (where $\Omega \subset \mathbb{R}$ for the 1 dimensional case):

$$\mathcal{W} \leq \sqrt{\mathcal{K}/2} \cdot \Omega \tag{4.30}$$

Which gives us:

Theorem 4.3: *Given two mechanisms A and B, we have:*

$$\epsilon_{A:B,i} \equiv \mathop{\mathbb{E}}_{\boldsymbol{v}} [\pi_{A,i}(\boldsymbol{v}) - \pi_{B,i}(\boldsymbol{v})] \leq \mathcal{W}(f_{\breve{\pi}_{A,i}}, f_{\breve{\pi}_{B,i}}) \leq \sqrt{\mathcal{K}(f_{\breve{\pi}_{A,i}}, f_{\breve{\pi}_{B,i}})/2} \cdot \overline{v_i} \tag{4.31}$$

where $\overline{v_i}$ is the maximum bid possible for the agent; and this holds for either ordering of the KL-divergence.

While useful theoretically, the constants in this bound are sufficiently large so as to not provide much insight as to the relative benefits of using the Wasserstein versus the KL-divergence, or some other similar metric. Still, we have thus bounded the difference in expected profit of two distinct mechanisms using a distribution metric.

4.7.3 Combining the Bounds

We can now put together the result we had from Theorem 4.1's equation (4.10) and Theorem 4.3's (4.31), giving us:

Theorem 4.4: *Given a strategyproof reference mechanism m^* and an individually rational reference-bounded mechanism m with identical outcome rule that is non-strategyproof for agent i, the expected ex ante gain in profit from unilateral deviation for agent i, in the sense of equation* (3.7)*, is bounded from above by the KL-divergence in agent i's profit distribution in mechanisms m^* and m. Formally:*

$$\epsilon_{EA,i} \equiv \mathop{\mathbb{E}}_{v_i}\left[\max_{v_i'}\left[\mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_i(v_i, v_i', \boldsymbol{v}_{\text{-}i})\right] - \mathop{\mathbb{E}}_{\boldsymbol{v}_{\text{-}i}}\left[\pi_i(\boldsymbol{v})\right]\right]\right] \leq \sqrt{\mathcal{K}(f_{\pi_i^*}, f_{\pi_i})/2} \cdot \overline{v_i} \tag{4.32}$$

Proof. Immediate from Theorem 4.1 and Theorem 4.3, as the upper bound on the former is equal to the lower bound on the latter when we substitute mechanisms m^* and m for mechanisms A and B. □

We have thus bounded the *ex ante* expected unilateral gain from deviation using a metric on the payoff distributions of a mechanism and its reference evaluated at truth. We consider this an interesting theoretical result, and one which lends support

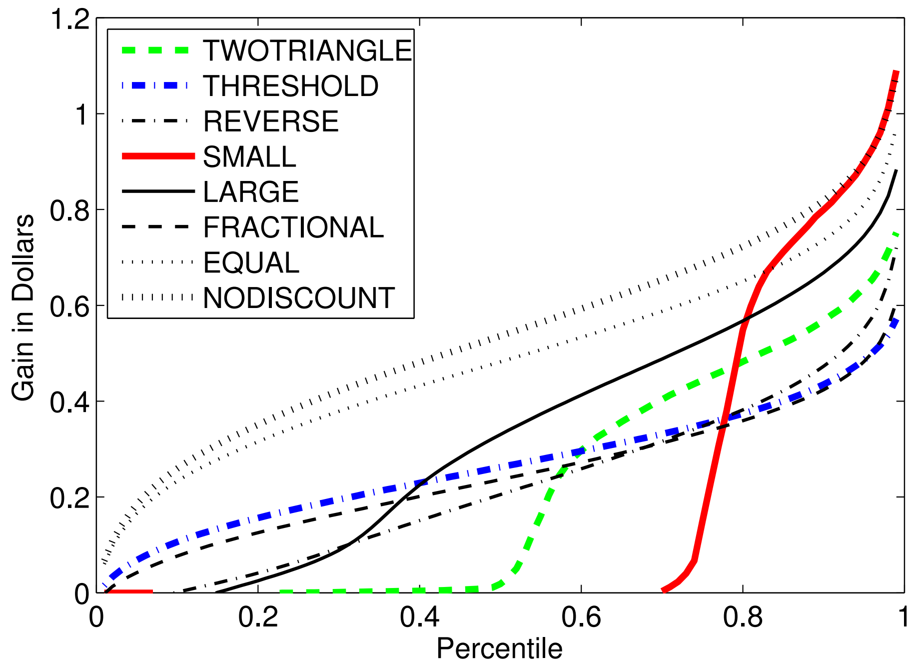


Figure 4.10: Quantiles of *ex post* gain in a CE with 10 goods and 10 agents and a uniform distribution on bundle value

to the idea that payoff distribution metrics can be useful in the analysis of incentives.

4.8 Quantiles of *ex post* Gain

So far in this chapter we have been advocating for a very simple metric based on the distribution of payoffs. In this section we argue that if the designer is willing to invest in some additional computation, he can reap rewards by instead considering quantiles of the *ex post* gains from deviation.

Figures 4.10 and Figure 4.11 show the quantiles of unilateral gain available to an agent in a full-fledged CE with 10 goods, 10 agents, and the *uniform* and *decay* value distributions respectively. The methodology for the creation of the graphs is similar to that in Section 3.7.2. If we focus on the worst case behavior (i.e. the 100$^{\text{th}}$

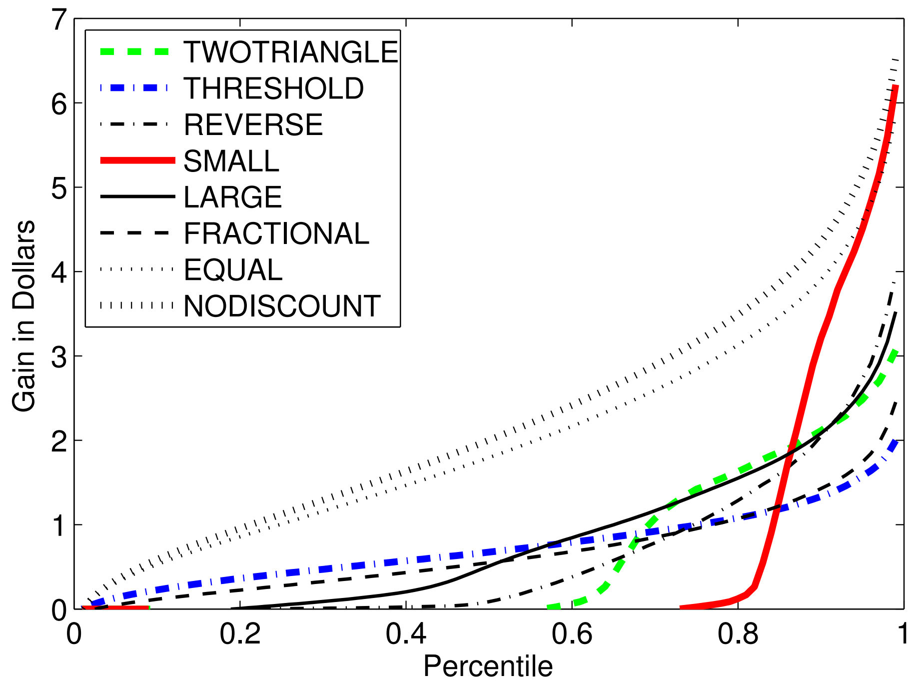


Figure 4.11: Quantiles of *ex post* gain in a CE with 10 goods and 10 agents and the *decay* distribution on bundle value

percentile), we can see that the Threshold rule is doing best. This is expected, as it minimizes the *ex post* maximal incentive to deviate. And, in fact, the rule does very well through at least the 90$^{\text{th}}$ percentile in both experiments. However, for the rest of the quantiles, the other rules do better, and often far better. We can see that the Small rule offers almost no incentive for deviation into the 70$^{\text{th}}$ percentile. So in the median case, one is clearly better off with this rule. And since the median and mean are always within one standard deviation of each other, the Small rule should also do very well in the expected case.

This expected case gain comes at a price however: the worst case behavior of Small is as bad as the case where we give out no discount at all (and given that Small gives no discount to those agents with the largest discount this behavior is expected too).

Thus we see a huge tradeoff between worst case and expected case optimal behavior. The novel TwoTriangle rule that will be described in Section 4.2.1 has reasonably robust behavior under both criteria, performing well in typical cases, and in the middle of the pack in the worst case.

This analysis gives us powerful information about the relative merits of the various rules under different optimality criteria. *Further, and intriguingly, the rule rankings that follow from the mid-quantiles of the analysis will turn out to match well the rankings we saw from a restricted equilibrium analysis in Table 4.2.* This indicates that by simply moving away from a worst case analysis (while still retaining *ex post* reasoning) we can already gain considerable traction in the analysis of strategic behavior.

This *ex post* criterion requires us to calculate optimal unilateral deviations when agents are given full information about other agents' value. For most mechanisms of interest this analysis can be made tractable through rudimentary analysis of the mechanism, e.g. by having the agent issue an optimal single-minded bid for the outcome they would truthfully win. Thus, the analysis does not require us to be at all sophisticated about the structural form of the equilibrium in order to readily produce the pertinent data.

This said, the analysis does require performing a search (and/or analysis), that while typically tractable, may nonetheless be undesirable to perform. Moreover, compelling as it may be, the analysis is limited to a unilateral deviation from truthful bidding, not a full Bayes-Nash equilibrium where all the agents simultaneously make their best-response deviation. Thus in this chapter, we seek a metric that requires only data available from historical instances directly, without requiring optimization to find counter-factual minimal regret bidding behavior. And we seek to analyze this metric in a close approximation to a BNE, not merely under unilateral deviations from the truth. We're interested in a measure that can well approximate this one, but only based on observed reports, e.g. in the equilibrium of the mechanism.

4.9 Related Work

In this section we will first show several existing methods from the literature for defining approximate strategyproofness. Then we will list some important domains where these notions are relevant. Finally, several existing approaches to automated design will be described.

4.9.1 Approximate Strategyproofness

A standard measure of approximate strategyproofness is *regret*, namely the loss in utility to an agent from reporting its true type compared to its best possible misreport, given reports of other agents. An *ϵ-strategyproof mechanism* is one in which truthful reporting achieves within $\epsilon > 0$ of the best possible utility, for all possible reports of other agents and all agent types. Schummer [2001] was the first to consider ϵ-strategyproof mechanisms and this approach was also considered by Kothari, Parkes, and Suri [2005] in the design of multi-unit auctions. This concept is meaningful when ϵ is small, for example smaller than the cost an agent incurs in reasoning about how to manipulate, because it is reasonable that agents will then behave truthfully.

This approach advocates *worst-case regret* as a metric of approximate strategyproofness, namely the worst-case loss in utility from behaving truthfully over all possible reports of other agents. But, as the maximal regret gets large it is not clear that regret provides the appropriate metric by which to quantify the degree of strategyproofness of a mechanism or guide mechanism design. If the setting is known to require mechanisms that have a certain amount of regret, we may forego significant opportunity to mitigate strategic behavior in the common cases, by targeting only the worst case. Another related notion of approximate strategyproofness is that of *strategyproof with high probability* [Archer, Papadimitriou, Talwar, and Tardos, 2004], which is similar in that it is well motivated only in places where the approximation bound can be made arbitrarily small. As we have seen, we instead want a criterion that captures a desire for the mechanism to be "as truthful as possible, as often as possible".

Budish [2009] recently advocated "strategyproofness in a large-market" as a criterion for selecting amongst two non-strategyproof mechanisms. This asks whether the mechanism will become strategyproof for a replica economy, in the limit as each agent becomes one of a continuum of agents with the same type. While a very useful design criterion, this does not by itself meet our needs of providing a metric with which to quantify approximate strategyproofness; rather, it is a binary classification.

Erdil and Klemperer [2010] suggest that mechanisms that have a low 'marginal incentive' for deviation will be highly strategyproof. Specifically, they suggest examining the derivative of the payoff function at the true report. When this derivative is small, perturbations in report are unlikely to result in a gain for the agent. Again, the concept is well motivated for small perturbations, but does not offer much help in analyzing settings where large profitable deviations may of necessity remain available to participants.

Our approach is also related to the work of Pathak and Sönmez [2009]. These authors propose a binary comparison between mechanisms, where mechanism ϕ is considered more manipulable than mechanism ψ if ϕ can be manipulated whenever ψ can, and in at least one context where it can't. This is a very general concept which does not rely on the existence of payments, and can thus be used by the authors to compare mechanisms for assignment problems without payments. However, the concept does not take into account either the degree of manipulability, or the probability of a given manipulation occurring. Thus while theoretically helpful in analyzing certain competing mechanisms, it does not provide design guidance for the complex mechanisms we wish to study here.

4.9.2 Over-Constrained Mechanism Design Settings

4.9.2.1 Budget Constrained Combinatorial Exchanges

One major motivation for the line of inquiry in this chapter is the desire to impose budget balance on payment rules used in the combinatorial exchange settings described in Section 2.7.2. For such a setting, the Vickrey-Clarke-Groves (VCG) mechanism provides the unique, strategyproof design [Holmström, 1979]. Thus because VCG may run at a deficit, no CE mechanism exists that is efficient, no deficit and individual rational [Myerson and Satterthwaite, 1983]. Moreover, no "second best" mechanism has been designed to date that maximizes expected efficiency by maximizing incentive compatibility while yet retaining individual-rationality and no deficit properties.

The notions of approximate incentive compatibility introduced in Chapter 3 are intended as a step in this direction; in this chapter we then applied these concepts to the domain explicitly, and constructed a metric that can be useful in the search for approximately incentive-compatible CEs. Chapter 5 will next introduce a full CE mechanism that can be built on top of any such rule choice.

4.9.2.2 Core Constrained Combinatorial Auctions

In addition to being concerned with the structure of the incentives presented to participants in a mechanism, we can also examine the extent to which the mechanism encourages subgroups of participants to defect from the dictated outcome and establish their own. As described in Section 2.9.1, payments that provide no impetus towards this type of defection are said to be in the **core**. The set of core payments may be empty for a combinatorial exchange setting. However, combinatorial auctions can support core payments. Consequently, there have been several recent papers on payment schemes for combinatorial auctions that impose core constraints.

Day and Raghavan [2007] argue that the bidder-Pareto-optimal core payments should be preferred within the core set, and offer a constraint generation technique for computing core payments using a linear program solver. Day and Milgrom [2008] offer additional theoretical analysis of minimal revenue core mechanisms in a full information setting, and draw connections with the matching mechanism literature. Day and Cramton [2008] note that the minimal revenue condition is not sufficient to uniquely specify payments. They propose to further minimize the L_2 distance to the VCG payments, and offer a quadratic program to do so. While the intuition that being close to the VCG payments seems reasonable, and parallels the distance metric approach described in Section 3.2.1, we have seen in Section 3.7.2 that it may not in fact result in the best incentive properties. These papers do, however, make a compelling case that the core constraints are very useful in mitigating the low revenue problem inherent in VCG payments, and that such mechanisms can be useful in practice: in particular, in recent UK bandwidth auctions.

Erdil and Klemperer [2010] argue for a payment rule that picks a vector in the minimum revenue core, but which is closest to a "reference" point, rather then to the Vickrey payments. The reference point for a given participant can be picked according to any function which does not depend on his bid. The specifics of this rule are provided in Section 3.7.1.1. Their analysis is restricted to a two-item setting, and we use it as a benchmark in Chapter 3 only; it has not been demonstrated in more general settings such as the one covered in this chapter.

4.9.2.3 Generalized Second Price Auctions

The Generalized Second Price (GSP) auctions described by Edelman et al. [2007] and used to sell on-line advertising at Google and Yahoo! also fit within this paradigm. Agents bid for slots, and those with the highest bids win the highest slots: i.e. the auction is cleared at the efficient trade. However, the price charged is that of the next lower slot, rather than the full VCG payment. This choice has a number of advantages: it is simpler than VCG (both computationally, and for participants to understand), it increases revenue over the VCG outcome, and importantly, the price charged tends to be locally insensitive to changes in report (low 'marginal incentive' for deviation). Further, Pathak and Sönmez [2009] use their binary relation between mechanisms to show that GSP is less manipulable than its first-price equivalent.

Slot auction design is another domain where strategic incentives are important but where we are constrained from using the VCG mechanism directly, in this case because we want both the mechanism and its bidding language to be simple. The techniques presented here and in the following chapter are notionally applicable to this settings as well, though the difficultly of defining a formal notion of 'simplicity' has led us to focus on other settings.

4.9.3 How to Automate Design

Several authors have considered mechanisms that deliberately choose inefficient outcomes in order to achieve strategyproofness. See early work by Myerson and Satterthwaite [1983], McAfee [1992a], and Barbera and Jackson [1995] for examples. However, the most pertinent work, described below, does choose the surplus-maximizing trade. In Chapter 3 we were interested in viewing the design problem as a form of constrained optimization. Several related approaches exist in the literature:

4.9.3.1 Minimizing Per Instance Regret

We can cast the rules described in Section 3.2.1 as a particular form of automated design. In particular Parkes et al. [2001a], define rules by minimizing the per instance distance between the payments chosen and those chosen by the VCG mechanism.

Different choices for the distance metric result in different rules but the net effect is to minimize regret. Because the VCG discount reflects the profit agents might achieve under an optimal misreport of the alternative mechanisms, the per instance criterion measures not only the distance to a strategyproof outcome, but also the potential gain of an optimal instance-specific misreport (i.e. if they had reported the critical value). However, we note that this is optimal only with respect to the *current instance*. The optimal misreport for one instance can easily cause the agent to lose the trade in other instances; thus, as we have shown, agents with less than full information will not adopt a strategy of attempting to report the critical value exactly. And thus minimizing the potential gain of such a misreport is the wrong goal in most settings.

With this said, the minimization of per-instance regret, which is the same criterion as ϵ-strategyproofness, is widely used in the literature due to its amenability to theoretical analysis and its direct interpretation.

Recent work by Lahaie along these lines leverages kernel methods from machine learning to construct approximate universal competitive equilibrium prices for a combinatorial auction [2010]. A particular constant for ϵ-strategyproofness is then derived in terms of properties of the kernel. This can be interpreted as the traditional measure of the strategyproofness of the mechanism. But intriguingly, it can also be viewed as using machine learning as a method for automating design by casting the optimal design problem as the search embedded in a machine learning kernel method. By searching to find the most separating prices the method is implicitly getting better strategic properties, in the per-instance sense, too.

4.9.3.2 Automated Mechanism Design

Conitzer and Sandholm [2002; 2003b] have proposed the paradigm of Automated Mechanism Design (AMD). In this approach, the complete mechanism is constructed as the result of an optimization problem. The system is fed the number of agents, and a prior distribution over their types. It then solves an optimization problem to find the non-manipulable mechanism with respect to some objective (such as social welfare). A succinct description of the AMD problem is provided by Guo and Conitzer [2010]. At its most fundamental, AMD formulates the design problem as the following optimization problem:

$$\begin{aligned}
&\operatorname*{argmax}_{m} \quad \sum_{\vec{\theta} \in \Theta^n} f_{\Theta^n}(\vec{\theta}) \sum_{i \in N} u_i(\theta_i, m(\vec{\theta})) && \text{Expected social welfare} \quad (4.33) \\
&\text{s.t.} \quad u_i(\theta_i, m(\vec{\theta})) \geq u_i(\theta_i, m(\theta_i'; \vec{\theta}_{\text{-}i})) && \text{Strategyproofness} \quad (4.34) \\
&\qquad\qquad \forall\, i \in N, \vec{\theta} \in \Theta^n, \theta_i' \in \Theta
\end{aligned}$$

where the optimization is over a mechanism $m : \Theta^n \to O$, a function from a vector of types to an outcome, f_{Θ^n} is a prior distribution over types and $u_i(\theta_i, o)$ gives the utility of an agent with a given type under a given outcome. If the domain and range of function m are finite, a randomized mechanism can be found by formulating the above as a linear program with m specified by a set of probability variables $p_m(o|\,\vec{\theta})\ \forall\ \vec{\theta} \in \Theta^n, o \in O$. If a deterministic mechanism is desired, then restricting the $p_m(\cdot)$ to be 0-1 variables yields the appropriate MIP.

Conitzer and Sandholm [2003a] implement AMD and solve several small design problems empirically showing automated design's applicability and scalability. Later [2004b], the authors consider the situation where the designer herself is self interested, but must obey the individual rationality constraints of participants. The authors also propose a computational framework [2007] that incrementally improves a mechanism, making it more and more strategyproof in each iteration. Conitzer's Ph.D. thesis [2006] includes longer descriptions of each of these methods. Likhodedov and Sandholm [2004a] formulate an AMD auction problem where they find the maximally efficient mechanism subject to the seller obtaining a required revenue (as well as the standard incentive constraints).

The authors consider the computational requirements of the optimization problem in several settings. In particular, they prove that if agents' preferences are quasi-linear and the objective is social welfare, then the mechanism design problem can be formulated as the above linear program and thus be polynomial if the joint type space is polynomial. And, by contrast, if payments are not allowed, then the problem becomes NP-hard. However, if the type space is exponential in the number of agents (as is typical), then the design problem will likewise become exponential because the formulation grows exponentially in both variables and constraints – even for a single item setting.

Several previous works have attempted to circumvent this computational complexity problem by restricting the AMD search space to a parametrized space of possible outcome and/or payment functions. Likhodedov and Sandholm [2004a;b; 2005] parametrize a VCG-style mechanism by considering all linear transformations of bidders' reports, then they optimize for maximal revenue over these linear factors. Guo and Conitzer [2009a;b] address the setting of *redistribution* mechanisms, where payments made by winning bidders must be redistributed to the participants. The authors consider both a class of linear redistribution mechanisms and a more general class of *discretized* mechanisms. They use AMD to search the parameter space of these classes to find mechanisms that redistribute the maximum value while maintaining strategyproofness, either in the worst case or in the expected case. Separately, they have considered the application of AMD to the problem of assigning m goods to two players without payments, defining a parametrized mechanism space over which to search with AMD [2010]. Constantin and Parkes [2007] consider a dynamic auction setting with interdependent values and use similar techniques to find the revenue-optimal mechanism over a restricted set of possible IC mechanisms. In all of these works, the search is over a parametrized space of mechanisms that is known to be strategyproof, meaning that the optimization formulation need not explicitly include incentive constraints. In Chapter 3, we do not restrict the space of mechanisms in this way, and thus must explicitly reason about incentive properties in our optimization. These existing approaches consider all possible trades, and gain tractability by looking at a restricted set of mechanisms. An important distinction is that we allow a very broad class of mechanisms, and instead limit the trades over which we optimize to those that are likely to occur in order to obtain computational tractability.

The work presented in Chapter 3 has much in common with AMD, in that it formulates the design problem as an optimization problem, and solves to find the best solution. It also shares the property of inherently requiring a prior (though this can of course be uniform if no knowledge is available). However, the work presented here differs in both the objectives and the constraints. Instead of requiring incentive compatibility and solving for the mechanism that best meets some goal (such as social welfare), we instead restrict the mechanism to a particular setting via constraints and then optimize to find the mechanism that is as close to incentive compatible as possible. For instance, in the examples presented in sections 3.6 and 3.7 we consider the class of mechanisms that chooses the efficient trade at reports for the winner determination method and then requires the payments to be in the core; other mechanism classes can easily be implemented instead. Problems of this form require a method for quantifying a notion of approximate incentive compatibility that is not in Automated Mechanism Design, as we considered in Section 3.3. In this chapter we took a different approach where we didn't need to reason directly about the joint type space and thus achieved far greater computational tractability.

4.9.3.3 Empirical Mechanism Design

Our work also relates to methods of *empirical mechanism design* [Vorobeychik, Kiekintveld, and Wellman, 2006; Vorobeychik, Reeves, and Wellman, 2007], in which one couples search through a parametrized mechanism space with an empirical methodology for solving the induced games. Approximate solutions to the full Bayes-Nash equilibrium are considered, the complexity of which limits the ease with which the method can be applied.

4.10 Summary

There are many important mechanism design settings that are over-constrained in the sense that we cannot simultaneously achieve strategyproofness while maintaining other desirable properties (e.g. budget balance). In these cases we may wish to

		Agents Have	
		Probabilistic Info.	**Full Information**
Designer Risk Tolerance	**Robustness**	Worst Case *Ex Ante* ⑥	Worst Case *Ex Post* ③
		Quantile *Ex Ante* ⑤	Quantile *Ex Post* ❷
	Performance	Expected *Ex Ante* ❹	Expected *Ex Post* ①

Table 4.6: Indirect approximate incentive compatibility conditions. Circled numbers indicate increasing computational cost. Filled circles are the two criteria we find most compelling.

enforce the satisfaction of these other properties, while designing mechanisms that are maximally *approximately strategyproof*.

4.10.1 Approximate Strategyproofness

In Section 3.3.1 we discussed how straightforward measures of approximate incentive compatibility, such as the distance agents manipulate in Bayes-Nash equilibrium, are problematic for the purposes of design. We have also argued that the standard measure, minimization of *regret* over all possible instances, is the wrong target when we know the regret to be of necessity quite substantial. For these cases, we instead proposed several criteria for approximate strategyproofness that we have argued are both more appropriate and more useful in design. We summarize these criteriia in Table 4.6, and note that the *ex ante* and quantile versions assume the designer has access to a prior on agent value.

The expected *ex ante* criterion is structured to well-match a Bayes-Nash equilibrium, even when evaluated at the true distribution and not the equilibrium distribution. However, *ex ante* conditions are extremely difficult to calculate as they require a stochastic optimization over a sampling of the joint value space in order to properly capture the information available to agents. By contrast *ex post* conditions are much easier to compute, and we believe that a quantile view of *ex post* data can be particularly informative at a modest computational cost.

In this analysis, we have been assuming that the designer has access to a prior distribution. If instead the designer does not have access to such information he can simply use a uniform distribution – which may be sufficient to provide reasonable fidelity in some cases. But the designer may instead choose to use the standard condition of minimizing regret in the worst possible instance across the entire joint value profile. This corresesponds the to the row labeled "robust" in the table. Note that the Worst Case *Ex Ante* condition as an appropriate choice for the designer to use when he doesn't have access to a prior himself, but he believes the participants will nonetheless be acting according to one. This forces the designer to adopt a Uniform prior for $f_{\boldsymbol{v}_{\text{-}i}}$ in his calculation of *ex ante* gain.

In the end, despite its simplicity, we believe that the regret condition focuses on such rare conditions that it is generally the wrong target. We believe that designers do generally have access to at least some form of prior, and that for complex markets it can be highly beneficial to use this information.

4.10.2 Computational Properties

Calculating even approximations to Bayes-Nash equilibria is extremely computationally expensive. Approaches to computational design that require such calculations, such as Empirical Mechanism Design, therefore have to limit both the complexity of the mechanisms and the scope of the agent strategies they consider. Instead, in Chapter 3 we proposed a constructive approach to designing approximately incentive-compatible mechanisms by formulating the problem as a type of constrained optimization. While of theoretical interest and practical applicability for small problems, and even with several approximations that we introduced, this method proves infeasible for large problems because of exponential growth of the formulation in the joint type space of the problem.

To gain traction in larger problem instances, we therefore adopted closed-form payment rules, and turned our attention to methods for comparing them. We can apply the same conditions for strategyproofness, now analytically instead of constructively. Unfortunately, though, *ex ante* conditions also prove computationally expensive, as they require calculating expectations over the joint type space. While this may be feasible with intelligent sampling approaches, we believe that by using a quantile analysis, in a far simpler-to-calculate *ex post* condition can, provide highly valuable information. While such methods do still require an optimization to find the best unilateral misreport, for many important domains such a search can be made with comparatively little effort by an appropriate linear or grid search algorithm.

However, even this computation may be too much for very large instances, or if we need a fast turnaround (e.g. in an online setting). In this chapter, we therefore considered simpler metrics that are based solely on the distribution of payoffs provided by the various rules. We evaluated these metrics both at equilibrium (as might be available from historical data) and at truth (as might be available via analytical analysis). We showed that despite their low computational burden, these measures can still provide useful information about the equilibrium properties of the mechanism being evaluated.

To summarize, we believe the expected case *ex ante* criterion is likely to be the proxy for the BNE with the highest fidelity. But it is also unattractive computationally. However, the quantiles of the *ex post* criteria are far easier to calculate, and provide highly informative about the incentive structure of a mechanism. We thus advocate for their use, where they are feasible to calculate. When even this computational burden is too high, we claim the KL-Divergence between the payoff distribution of the mechanism being evaluated and a strategyproof reference mechanism is both computationally simple, and an effective predictor of equilibrium behavior.

4.10.3 Quantification via the KL-Divergence

The particular measure of equilibrium behavior that we advocate when computation is at a premium is a form of the KL-divergence. Specifically, it is one defined based on the difference between a distribution on agent payoffs in a mechanism and that under a reference, strategyproof mechanism, both evaluated with respect to the true distribution on agent valuations. This metric is shown to be more informative, in terms of correlating with the deviation from truthful bidding in equilibrium, than other regret-based metrics. In the context of CEs, our results establish that by seeking to match the payoffs in a reference mechanism in distribution, a mechanism designer can achieve a mechanism that is maximally strategyproof in the sense of minimizing the amount by which agents will deviate from truthful bidding in equilibrium with only historical (or analytic) data and minimal computation.

4.10.4 Observations Regarding Payment Rules

We observe that minimizing maximal *ex post* regret (given truthful bids) does not necessarily lead to optimal designs; e.g., the Threshold mechanism is designed this way, but the Small mechanism generates a better (closer to truthful) equilibrium while also minimizing average regret in equilibrium. Despite its strong equilibrium behavior, we point out that the Small rule may be undesirable from the perspective of the zero profit it assigns to precisely those agents who would nominally be destined to receive the most. Accordingly we propose a new rule, TwoTriangle, which is an affine mixture of the Small and Threshold rule, and which appears to have good equilibrium properties like Small, but which provides some profit margin to the 'larger' winners like Threshold.

In the end, the best rule will need to be determined by the specifics of the domain: how informed the participating agents are, the risk tolerance of the designer when it comes to the incentives he's providing, and most importantly, the structure of the type space itself. We have provided several tools for evaluating those influences empirically, using data pertinent to the environemts where we seek a design.

I would not give a fig for the simplicity this side of complexity, but I would give my life for the simplicity on the other side of complexity.

– Oliver Wendell Holmes
Supreme Court Justice, 1902–1932

5

ICE: An Iterative Combinatorial Exchange

5.1 Motivation

This chapter 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 [Compte and Jehiel, 2007; Sandholm and Boutilier, 2006]. 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.[1]

- ICE is a hybrid between a demand-revealing process and a direct-revelation mechanism, with simple (linear) prices guiding preference elicitation, but with bids submitted through direct claims about valuation functions in the *TBBL* language, and these expressive bids used ultimately 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., 2001a] in defining final payments, which minimizes the *ex post* regret for truthful bidding across all

[1]See early work by Parkes [1999; 2005] for the use of bounds in single item auctions, and Hudson and Sandholm [2004] for application to CAs. Here we use bounds in the context of a fully expressive CE.

budget-balanced payment rules, when holding the bids from other participants fixed; any of the rules discussed in the previous two chapters could be used instead. The choice of payment rule is orthogonal to the design of ICE and thus not the focus here. However, 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 who 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 and 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. Additionally, we provide benchmarks that enable both a quantitative and 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.

Outline

Section 5.2 defines a sealed-bid CE, introducing *TBBL* and providing the MIP that is used to solve winner determination. Section 5.3 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 5.4 gives a number of illustrative examples of the operation of ICE. Section 5.5 presents our main experimental results. Works related to this topic are discussed in Section 5.6. We summarize the results in Section 5.7.

5.2 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.

5.2.1 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 accomodate bidders who are both buying and selling goods simultaneously (i.e., bidders carrying on transactions ranging from 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). 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. *TBBL* is designed such that these changes in value are expressed on *trades* rather than with respect to the total value of allocations. Examples will shortly be provided in Figures 5.1 and 5.2, and discussed below.

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 $\mathit{Leaf}(T_i) \subseteq T_i$ be the subset of nodes representing the leaves of T_i and let $\mathit{Child}(\beta) \subseteq T_i$ denote the children of node β. All nodes except leaves are labeled with the *interval-choose* operator $\mathrm{IC}_x^y(\beta)$. 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 $\mathrm{IC}_x^y(\beta)$ 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; $\mathrm{IC}_1^n(\beta)$ 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 $\mathrm{IC}_x^y(\beta)$ node is defined by the following two rules:

R1 Node β with $\mathrm{IC}_x^y(\beta)$ 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 $\mathrm{IC}_x^y(\beta)$:

$$x \ sat_i(\beta) \leq \sum_{\beta' \in Child(\beta)} sat_i(\beta') \leq y \ sat_i(\beta) \tag{5.1}$$

Equation (5.1) enforces the *interval-choose* constraints, by ensuring that no more

[2]This equivalence implies that *TBBL* can directly express the XOR, OR and XOR/OR languages [Nisan, 2006; Sandholm, 2002a;b].

[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.

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:

$$\sum_{\beta \in Leaf(T_i)} q_i(\beta, j) sat_i(\beta) \leq \lambda_{ij}, \quad \forall j \in G \tag{5.2}$$

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 as sold in the tree be 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:

$$v_i(T_i, \lambda_i) = \max_{sat_i} \sum_{\beta \in T_i} v_i(\beta) sat_i(\beta) \tag{5.3}$$

$$\text{s.t.} \quad (5.1), \; (5.2)$$

Example 5.1: *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 5.1 shows a* TBBL *valuation tree for expressing this kind of swap.*

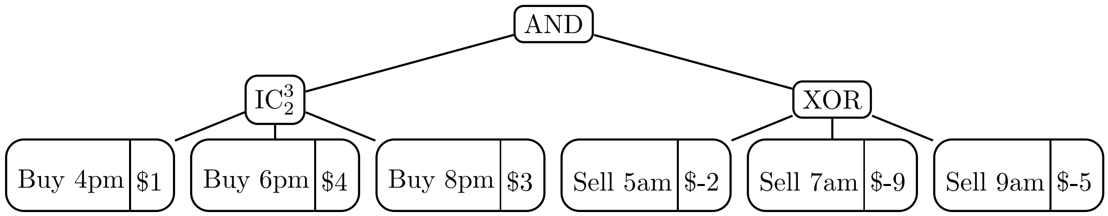


Figure 5.1: A simple *TBBL* tree for an airline interested in trading landing slots.

5.2.2 Winner Determination

The problem of determining an efficient trade given bids is called the *winner determination* (WD) problem (see Section 2.7.3. The WD problem in CAs (and thus also in CEs) is NP-hard [Rothkopf, Pekeč, and Harstad, 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 and Wolsey, 1999]. A similar approach has proved successful for solving the WD problem in CAs [Boutilier, 2002; de Vries and Vohra, 2003; 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 a *TBBL* specific form of the WD problem for bid trees $T = (T_1, \ldots, T_n)$ and initial allocation x^0:

$$\begin{aligned}
\text{WD}(T, x^0): \ & \max_{\lambda, sat} \sum_i \sum_{\beta \in T_i} v_i(\beta) sat_i(\beta) \\
\text{s.t.} \quad & (2.14), (2.15) \\
& sat_i \in valid(T_i, \lambda_i), \qquad \forall i \\
& sat_i(\beta) \in \{0, 1\}, \lambda_{ij} \in \mathbb{Z},
\end{aligned}$$

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 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.

5.2.3 Payments

If we are willing to run our CE at a potential budget deficit, we can use the VCG payment scheme covered in Section 2.7.4. However, if we need budget balance we will need to choose another rule, such as those proposed by Parkes et al. [2001a], and described in Section 3.2.1 and analyzed in Chapter 4. Here, by way of example, we adopt the Threshold rule. To reiterate, this rule charges:

$$p_{\text{thresh},i} = \hat{v}_i(\lambda_i^*) - \Delta_{\text{thresh},i}, \tag{5.4}$$

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 *per instance* from the VCG outcome across all balanced rules.

Example 5.2: *Consider the two bidders in Figure 5.2. 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 in 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*

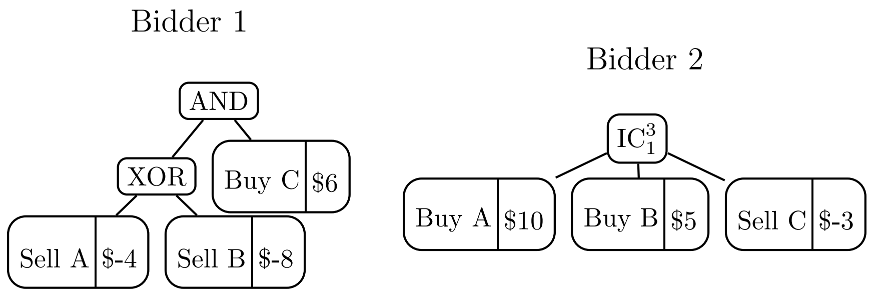


Figure 5.2: Two bidders and three items $\{A, B, C\}$. The efficient trade is for bidder 1 to sell A and buy C.

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.

5.3 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 annotates a single *TBBL* tree with upper and lower bounds on his valuation. The ICE mechanism then proceeds in rounds, as illustrated in Figure 5.3.

ICE is a *proxied design* in which each bidder has a proxy to facilitate his valuation

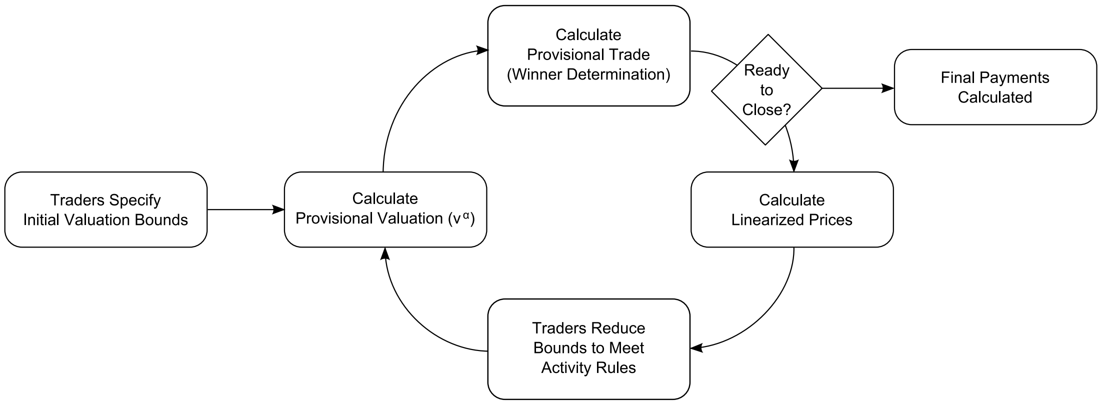


Figure 5.3: ICE system overview

refinement. In each round, a bidder responds to prices by 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 of these lower bounds guarantee that a bidder will be willing to pay at least this amount (or receive a payment of this amount) in order to complete the trade.

5.3.1 An ICE Round

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 let us adopt $\mathrm{WD}(v)$ 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 WD($\underline{v}$)) is close to the total value from the efficient trade λ^*_i:

$$\mathrm{EFF}(\underline{\lambda}) = \frac{\sum_i v_i(\underline{\lambda}_i)}{\sum_i v_i(\lambda^*_i)} = \frac{v(\underline{\lambda})}{v(\lambda^*)} \geq \Delta^* \tag{5.5}$$

However, the true valuation v and thus the trade λ^* are uncertain within ICE and therefore we will later introduce techniques to establish 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 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 $\omega^{\mathrm{eff}} \geq \Delta^*$ 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 $v^\alpha = (v^\alpha_1, \ldots, v^\alpha_n)$, where $v^\alpha_i(\lambda_i) = \alpha \underline{v}_i(\lambda_i) + (1-\alpha)\overline{v}_i(\lambda_i)$, expressed with a *TBBL* tree in which the value on node $\beta \in T_i$ is $v^\alpha_i(\beta) = \alpha \underline{v}_i(\beta) + (1-\alpha)\overline{v}_i(\beta)$.

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, $\phi \in \mathbb{R}^m_{\geq 0}$, 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, ϕ) to each bidder $i \in N$, and whether 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, and (d) offer 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.

5.3.2 Extending *TBBL* to Allow Upper and Lower Bounds

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 (5.3). The bounds on a trade can be interpreted as limmiting 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

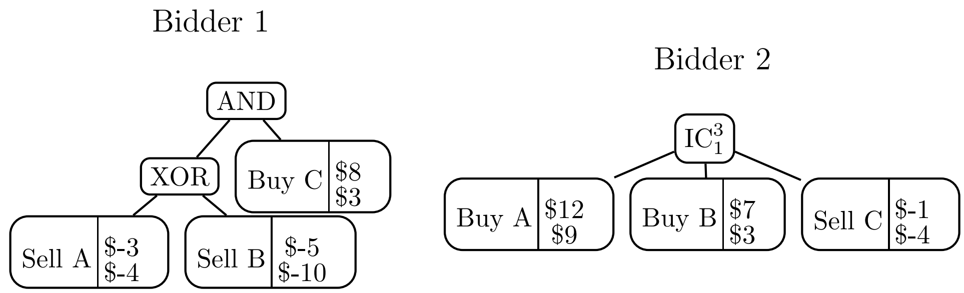


Figure 5.4: Two bidders, each with partial value information defined on their bid tree. One can already prove that the efficient trade is for bidder 1 to sell A and buy C.

$\underline{v}_i(T_i, \lambda_i) \leq \overline{v}_i(T_i, \lambda_i)$ 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 5.2:

Example 5.3: *The structure of the bidders' trees in Figure 5.4 is the same as in Example 5.2 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.*

5.3.3 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, 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 these 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.

5.3.3.1 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 in the clock-proxy auction in a one-sided CA [Ausubel et al., 2006].

[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.

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 be enough information in valuation bounds to establish that one trade is weakly preferred to all other trades at the prices, i.e.[7]

$$\exists \breve{\lambda}_i \in \mathcal{F}_i(x^0) \;\text{ s.t. }\; v_i'(\breve{\lambda}_i) - p^{\phi}(\breve{\lambda}_i) \geq v_i'(\lambda_i') - p^{\phi}(\lambda_i'), \quad \forall v_i' \in T_i, \forall \lambda_i' \in \mathcal{F}_i(x^0) \tag{RPAR}$$

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 $\breve{\lambda}_i \in \text{argmax}_{\lambda_i \in \mathcal{F}_i(x^0)}[\hat{v}_i(\lambda_i) - p^{\phi}(\lambda_i)]$ satisfies RPAR. We say that prices ϕ are *strict EQ prices* for $(v^{\alpha}, \lambda^{\alpha})$ when:

$$v_i^{\alpha}(\lambda_i^{\alpha}) - p^{\phi}(\lambda_i^{\alpha}) > v_i^{\alpha}(\lambda_i') - p^{\phi}(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0) \setminus \{\lambda_i^{\alpha}\}, \tag{5.6}$$

for every bidder $i \in N$.

Theorem 5.1: *If prices ϕ 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:

$$v_i^{\alpha}(\breve{\lambda}_i) - p^{\phi}(\breve{\lambda}_i) \geq v_i^{\alpha}(\lambda_i') - p^{\phi}(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0). \tag{5.7}$$

[7] For consistancy with notation need in describing the pricing rule, we denote the payment made under linear prices ϕ as p^{ϕ}

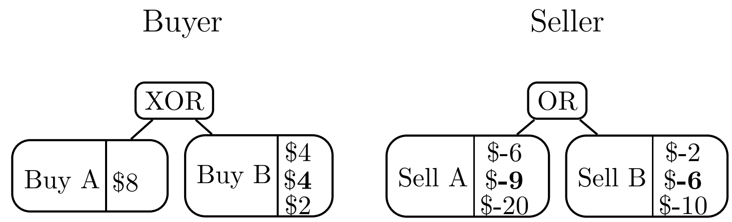


Figure 5.5: An example to illustrate the failure of the simple RPAR rule without strict EQ prices. True values are shown in **bold** and are such that the efficient outcome is no trade.

Moreover, we must have $\breve{\lambda}_i = \lambda_i^\alpha$, because $v_i^\alpha(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha) > v_i^\alpha(\lambda_i') - p^\phi(\lambda_i')$ 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:

$$v_i(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha) \geq v_i(\lambda_i') - p^\phi(\lambda_i'), \quad \forall \lambda_i' \in \mathcal{F}_i(x^0), \tag{5.8}$$

from which prices p^ϕ are EQ prices with respect to true valuations. The efficiency claim then follows from the welfare theorem, Theorem 2.7. □

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, as illustrated in the following example:

Example 5.4: *In the* TBBL *trees shown in Figure 5.5 no trade will 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 $\phi = (6, 2)$ 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

$$\theta_i^\phi(\breve{\lambda}_i, \lambda_i', v_i') \geq 0, \quad \forall v_i' \in T_i, \forall \lambda_i' \in \mathcal{F}_i(x^0) \tag{5.9}$$

$$\textbf{and}\text{ either } \breve{\lambda}_i = \lambda_i^\alpha \quad \textit{or} \;\; \theta_i^\phi(\breve{\lambda}_i, \lambda_i^\alpha, v_i') > 0,\ \forall v_i' \in T_i. \tag{5.10}$$

where $\theta_i^\phi(\lambda_i, \lambda_i', v_i') = v_i'(\lambda_i) - p^\phi(\lambda_i) - (v_i'(\lambda_i') - p^\phi(\lambda_i'))$ denotes the increase in profit to bidder i for trade λ_i over λ_i' given v_i' and prices ϕ. (5.9) 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 5.4. 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 5.1: *Given provisional trade λ^α, linear prices ϕ, 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:*

$$\theta_i^{\phi}(\lambda_i^{\alpha}, \lambda_i', v_i') \geq -\delta, \quad \forall v_i' \in T_i, \forall \lambda_i' \in \mathcal{F}_i \tag{5.11}$$

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

$$\theta_i^{\phi}(\breve{\lambda}_i, \lambda_i', v_i') \geq 0, \quad \forall v_i' \in T_i, \forall \lambda_i' \in \mathcal{F}_i(x^0) \tag{5.12}$$

$$\theta_i^{\phi}(\breve{\lambda}_i, \lambda_i^{\alpha}, v_i') > \delta, \quad \forall v_i' \in T_i \tag{5.13}$$

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* (5.11), *or some other trade satisfies RPAR* (5.12) *and is strictly preferred [by at least δ] to the provisional trade* (5.13). Just as for RPAR, one can show that a bidder can always meet δ-MRPAR (for any δ) by defining an exact valuation.[8]

In analyzing the properties of the δ-MRPAR rule, the following lemma will be useful:

Lemma 5.2: *If every bidder i meets δ-MRPAR without precluding v_i^{α} from his updated bid tree, and prices ϕ 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.*

[8]Let v_i denote this valuation. In the case where δ-MRPAR is not satisfied via (5.11) then $\breve{\lambda}_i \in \mathrm{argmax}_{\lambda_i \in \mathcal{F}_i(x^0)}[v_i(\lambda_i) - p^{\phi}(\lambda)]$ will satisfy δ-MRPAR as follows: It satisfies (5.12) by construction. Now, let λ_i' denote the trade with $v_i(\lambda_i') - p^{\phi}(\lambda_i') > v_i(\lambda_i^{\alpha}) - p^{\phi}(\lambda_i^{\alpha}) + \delta$. We have $v_i(\breve{\lambda}_i) - p^{\phi}(\breve{\lambda}_i) \geq v_i(\lambda_i') - p^{\phi}(\lambda_i') > v_i(\lambda_i^{\alpha}) - p^{\phi}(\lambda_i^{\alpha}) + \delta$, and (5.13).

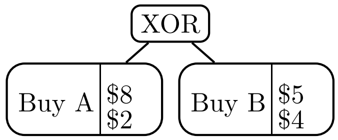
Proof. Fix bidder i. By δ-EQ, we have $\theta_i^\phi(\lambda_i^\alpha, \lambda_i', v_i^\alpha) \geq -\delta$ for all $\lambda_i' \in \mathcal{F}_i(x^0)$. Consider any $\breve{\lambda}_i \neq \lambda_i^\alpha$. Because v_i^α remains in the bid tree, we must have $\theta_i^\phi(\breve{\lambda}_i, \lambda_i^\alpha, v_i^\alpha) \leq \delta$ and δ-MRPAR cannot be satisfied via (5.12) and (5.13). Therefore, δ-MRPAR is satisfied for every bidder via (5.11) and with provisional trade λ^α the satisfying trade. Therefore we prove that prices, ϕ, are δ-approximate EQ prices for all valuations, including the true valuation, since bidders are straightforward and this is within their bounds. The efficiency of the trade follows from Theorem 2.8from Section 2.10.3. □

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.

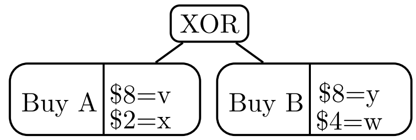
Theorem 5.3: *ICE with δ-MRPAR is $2\min(M, \frac{n}{2})\delta$-efficient when prices are δ-accurate with respect to $(v^\alpha, \lambda^\alpha)$ upon termination and bidders are straightforward.*

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 5.2, 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



(a) Passes δ-MRPAR



(b) Fails δ-MRPAR

Figure 5.6: δ-MRPAR where the provisional trade is "Buy A", $\phi_A = 3, \phi_B = 4$ and $\delta = 2$

for a domain in which approximately accurate EQ prices exist:

Corollary 5.4: *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.

Example 5.5: *To illustrate the δ-MRPAR rule consider a single bidder with a valuation tree as in Figure 5.6a. Suppose the provisional trade λ_i^α allocates A to the bidder, and with prices $\phi_A = 3, \phi_B = 4$ 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 5.6b, 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 5.6a, 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* $v = \$3, w = \$6$*.*

5.3.3.2 MRPAR Computation and Bidder Feedback

The definition of MRPAR by a naive interpretation 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. In this section we present a method for checking MRPAR given prices ϕ, 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. The details of how our agents use this information are described in Section 5.5.1.

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:

$$\lambda_i^L \in \operatorname*{argmax}_{\lambda_i \in \mathcal{F}_i(x^0)} \underline{v}_i(\lambda_i) - p^\phi(\lambda_i) \tag{5.14}$$

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 $\underline{v}_i(\lambda_i) - p^\phi(\lambda_i)$, 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.

Next, we will find it useful to define the ***perturbed valuation with respect to a trade*** λ_i on a bid tree T_i, by assigning the following values to each node β:

$$\tilde{v}_i(\beta) = \begin{cases} \underline{v}_i(\beta) & \text{, if } \beta \in sat_i(\lambda_i) \\ \overline{v}_i(\beta) & \text{, otherwise,} \end{cases} \tag{5.15}$$

where $\beta \in sat_i(\lambda_i)$ if and only if node β is satisfied in tree T_i at the lower bound valuations $\underline{v}_i$ on nodes for the trade λ_i. This valuation function, $\tilde{v}_i$, is minimizes the value on nodes satisfied by trade λ_i and maximizes the value on other nodes.

Now, given perturbed valuation $\tilde{v}_i$, defined with respect to trade λ_i^L (as in Section 5.3.2), we can define a *witness trade*, λ_i^U, as:

$$\lambda_i^U \in \arg \max_{\lambda_i \in \mathcal{F}_i(x^0)} \tilde{v}_i(\lambda_i) - p^\phi(\lambda_i). \tag{5.16}$$

This can be found by solving a third MIP. Given prices ϕ, 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) $\underline{v}_i(\lambda_i^L) - p^\phi(\lambda_i^L) \geq \tilde{v}_i(\lambda_i^U) - p^\phi(\lambda_i^U)$,**and**

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

We now establish that C-MRPAR is equivalent to MRPAR, as defined by (5.11)–(5.13).

Lemma 5.5: *Given trades λ_i and λ_i', prices ϕ, and tree T_i, we have $\theta_i^\phi(\lambda_i, \lambda_i', v_i') \geq 0$, $\forall v_i' \in T_i$ if and only if $\underline{v}_i(\lambda_i) - p^\phi(\lambda_i) \geq \tilde{v}_i(\lambda_i') - p^\phi(\lambda_i')$, 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 $\underline{v}_i(\lambda_i) - p^\phi(\lambda_i) \geq \tilde{v}_i(\lambda_i') - p^\phi(\lambda_i')$ but there exists some $v_i' \in T_i$ such that $v_i'(\lambda_i) - p^\phi(\lambda_i) < v_i'(\lambda_i') - p^\phi(\lambda_i')$. 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

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

$$<$$

$$\sum_{\beta \in \lambda_i' \setminus \lambda_i} v_i'(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} v_i'(\beta) - \sum_{\beta \in \lambda_i \cap \lambda_i'} v_i'(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} \underline{v}_i(\beta) - p^\phi(\lambda_i') \quad (5.17)$$

$$\Rightarrow \quad \sum_{\beta \in \lambda_i \setminus \lambda_i'} \underline{v}_i(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} \underline{v}_i(\beta) - p^\phi(\lambda_i) < \sum_{\beta \in \lambda_i' \setminus \lambda_i} \overline{v}_i(\beta) + \sum_{\beta \in \lambda_i \cap \lambda_i'} \underline{v}_i(\beta) - p^\phi(\lambda_i') \quad (5.18)$$

$$\Rightarrow \quad \underline{v}_i(\lambda_i) - p^\phi(\lambda_i) < \tilde{v}_i(\lambda_i') - p^\phi(\lambda_i') \quad (5.19)$$

which is a contradiction. $\square$

Lemma 5.6: *Given trade λ_i, prices ϕ, and tree T_i then $\theta_i^{\phi}(\lambda_i, \lambda'_i, v'_i) \geq 0$, $\forall v'_i \in T_i$, $\forall \lambda'_i \in \mathcal{F}_i(x^0)$, if and only if $\underline{v}_i(\lambda_i) - p^{\phi}(\lambda_i) \geq \tilde{v}_i(\lambda_i^U) - p^{\phi}(\lambda_i^U)$, 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 $\underline{v}_i(\lambda_i) - p^{\phi}(\lambda_i) \geq \tilde{v}_i(\lambda_i^U) - p^{\phi}(\lambda_i^U)$ but there exists some $\lambda'_i \in \mathcal{F}_i(x^0)$ and $v'_i \in T_i$ such that $\theta_i^{\phi}(\lambda_i, \lambda'_i, v'_i) < 0$. By Lemma 5.5, this means $\underline{v}_i(\lambda_i) - p^{\phi}(\lambda_i) < \tilde{v}_i(\lambda'_i) - p^{\phi}(\lambda'_i)$. But, we have a contradiction because

$$\underline{v}_i(\lambda_i) - p^{\phi}(\lambda_i) \geq \tilde{v}_i(\lambda_i^U) - p^{\phi}(\lambda_i^U) \tag{5.20}$$

$$= \max_{\lambda''_i \in \mathcal{F}_i(x^0)} \tilde{v}_i(\lambda''_i) - p^{\phi}(\lambda''_i) \geq \tilde{v}_i(\lambda'_i) - p^{\phi}(\lambda'_i) \tag{5.21}$$

□

Theorem 5.7: *C-MRPAR is equivalent to δ-MRPAR for $\delta = 0$.*

Proof. Comparing (5.9) and (5.10) with C-MRPAR, and given Lemmas 5.5 and 5.6, 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 $\max_{\lambda_i \in \mathcal{F}_i(x^0)}[\underline{v}_i(\lambda_i) - p^{\phi}(\lambda_i)]$. Otherwise, there is some λ'_i with $\underline{v}_i(\lambda'_i) - p^{\phi}(\lambda'_i) > \underline{v}_i(\breve{\lambda}_i) - p^{\phi}(\breve{\lambda}_i)$. A contradiction with (5.9).

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 $\theta_i^{\phi}(\lambda'_i, \breve{\lambda}_i, \overline{v}_i) >$

$\theta_i^\phi(\lambda_i', \breve{\lambda}_i, \underline{v}_i)$. But, since λ_i' has the same profit as $\breve{\lambda}_i$ at $\underline{v}_i$ we have $\theta_i^\phi(\lambda_i', \breve{\lambda}_i, \underline{v}_i) = 0$ and so $\theta_i^\phi(\lambda_i', \breve{\lambda}_i, \overline{v}_i) > 0$. This is a contradiction with (5.9).

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 (5.14) 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 $\theta_i^\phi(\breve{\lambda}_i, \lambda_i', v_i') \geq 0$, $\forall v_i' \in T_i$. In particular, $\tilde{v}_i(\breve{\lambda}_i) - p^\phi(\breve{\lambda}_i) \geq \tilde{v}_i(\lambda_i') - p^\phi(\lambda_i')$, where $\tilde{v}_i$ is defined with respect to $\breve{\lambda}_i$, and equivalently,

$$\underline{v}_i(\breve{\lambda}_i) - p^\phi(\breve{\lambda}_i) \geq \tilde{v}_i(\lambda_i') - p^\phi(\lambda_i'). \tag{5.22}$$

On the other hand,

$$\underline{v}_i(\breve{\lambda}_i) - p^\phi(\breve{\lambda}_i) = \underline{v}_i(\lambda_i') - p^\phi(\lambda_i'), \tag{5.23}$$

since both are in Λ_i^L. Taking (5.22) together with (5.23), 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 (5.23) 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 5.6: *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 5.7: *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* $\underline{v}_i(\lambda_i^L) - p^\phi(\lambda_i^L) = 5 + 1 - 3 = 3$*. 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 $\theta_i^{\phi}(\lambda_i^{\alpha}, \lambda_i', v_i') \geq -\delta$ for all $v_i' \in T_i$, all $\lambda_i' \in \mathcal{F}_i(x^0)$ directly, by application of Lemma 5.6 with valuation $\tilde{v}_i$ defined with respect to trade λ_i^{α}, and test

$$\underline{v}_i(\lambda_i^{\alpha}) - p^{\phi}(\lambda_i^{\alpha}) \geq \tilde{v}_i(\lambda_i^U) - p^{\phi}(\lambda_i^U) - \delta \tag{5.24}$$

(2) If this is not satisfied then fall back on C-MRPAR to verify (5.12) and (5.13), with candidate passing trade λ_i^L modified from (5.14) to drop tie-breaking in favor of λ_i^{α} and with the second step of C-MRPAR modified to require $\underline{v}_i(\lambda_i^L) - p^{\phi}(\lambda_i^L) > \tilde{v}_i(\lambda_i^{\alpha}) - p^{\phi}(\lambda_i^{\alpha}) + \delta$, again with $\tilde{v}_i$ defined with respect to λ_i^L.

The argument adopted in the proof of Theorem 5.7 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.

5.3.3.3 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 5.3.5, 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 5.5.[9]

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:

[9]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 5.8: *If there is no value uncertainty on any interesting nodes in the bid trees of any bidders, and bidders are straightforward, then λ^α is efficient.*

Proof. An absense of value uncertainty and thus the presence of 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. □

DIAR focuses a bidder on interesting nodes associated with 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 ϕ 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 ϕ might misprice some trade $\lambda_i^k \in \mathcal{F}_i(x^0)$ with respect to bidder i's true valuation:[10]

$$\overline{\delta}_i^k = \max_{v_i' \in T_i}[v_i'(\lambda_i^k) - p^\phi(\lambda_i^k) - (v_i'(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha))] \tag{5.25}$$

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

[10] Related ideas for preference elicitation in CEs can be found in the early the work of Smith, Sandholm, and Simmons [2002].

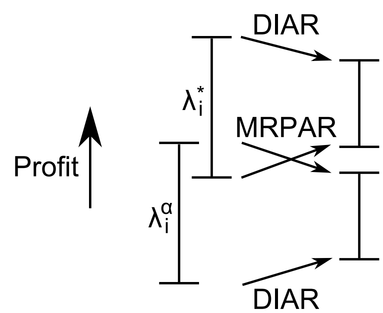


Figure 5.7: Stylized effect of MRPAR and DIAR on the bounds of the λ_i^α and λ_i^* trades

$\delta_i = \max_{\lambda'_i \in \mathcal{F}_i(x^0)}[v_i^\alpha(\lambda'_i) - p^\phi(\lambda'_i) - (v_i^\alpha(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha))]$ 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 2.8 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 ϕ 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 5.7 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)

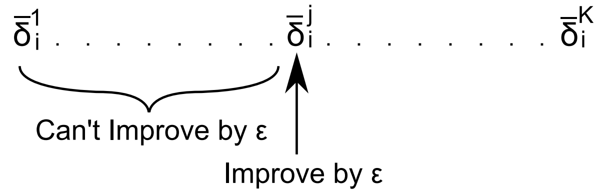


Figure 5.8: Bidder i's trades, ordered such that DIAR error decreases from left to right. The bidder must reduce, by at least ϵ, the DIAR error on the trade with the greatest error for which such a reduction is possible and prove (via valuation bounds) that it is impossible to improve by ϵ any trades with larger error.

is large. The rule is illustrated in Figure 5.8. DIAR is parameterized by some $\epsilon \geq 0$. We refer to the formal rule as ϵ-DIAR:

Definition 5.2: *To satisfy ϵ-DIAR given provisional trade λ_i^α and prices ϕ, the bidder must modify his valuation bounds to:*

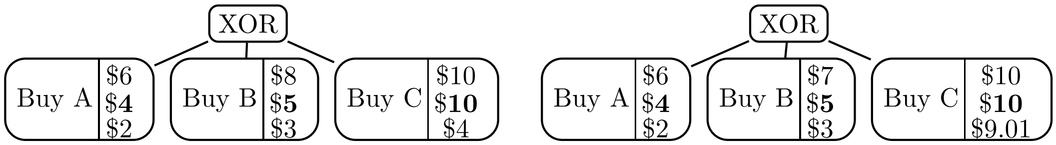
(a) reduce the DIAR error on some trade, $\lambda_i^j \in \mathcal{F}_i(x^0)$, by at least ϵ **and**

(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 $\lambda_i^k \in \mathcal{F}_i(x^0)$.*

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.

Example 5.8: *Consider the tree in Figure 5.9a when the provisional trade is "buy A", prices $\phi = (\$4, \$5, \$6)$ and DIAR parameter $\epsilon = 1$. The DIAR error on each*



(a) Fails DIAR **(b)** Passes DIAR

Figure 5.9: Respecting DIAR where the provisional trade is "Buy A", $\phi_A = 4, \phi_B = 5, \phi_C = 6$ and $\epsilon = 1$

trade, defined via (5.25)*, and listed in decreasing order, are:*

$$C \rightarrow \overline{\delta}^1 = (\$10 - \$6) - (-\$2) = \$6$$

$$B \rightarrow \overline{\delta}^2 = (\$8 - \$5) - (-\$2) = \$5$$

$$\emptyset \rightarrow \overline{\delta}^3 = (\$0 - \$0) - (-\$2) = \$2$$

$$A \rightarrow \overline{\delta}^4 = (\$2 - \$4) - (-\$2) = \$0,$$

where $-\$2 = \$2 - \$4$ *is the worst-case profit from the provisional trade. Now, we see that* $\overline{\delta}^1$ *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 5.9b.*

Lemma 5.9: *When ICE incorporates DIAR, a straightforward bidder must eventually reveal complete value information on all interesting nodes in his bid tree as* $\epsilon \rightarrow 0$*.*

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. $\square$

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

Theorem 5.10: *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 5.8 and Lemma 5.9. □

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 5.3.5).

5.3.3.4 DIAR Computation and Bidder Feedback

In this section we present a method to check ϵ-DIAR given prices ϕ, 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. The details of how our agents use this information are described in Section 5.5.1.

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:

$$\Delta_i^P = \max_{\lambda_i \in \mathcal{F}_i(x^0)} [\tilde{v}_i^0(\lambda_i) - p^\phi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha))] \tag{5.26}$$

$$\text{s.t.} \quad (\tilde{v}_i^0(\lambda_i) - p^\phi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha)))$$

$$- (\tilde{v}_i^1(\lambda_i) - p^\phi(\lambda_i) - (\underline{v}_i^1(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha))) \geq \epsilon \tag{5.27}$$

$$= -C + \max_{\lambda_i \in \mathcal{F}_i(x^0)} \tilde{v}_i^0(\lambda_i) - p^\phi(\lambda_i) \tag{5.28}$$

$$\text{s.t.} \quad \tilde{v}_i^0(\lambda_i) - \underline{v}_i^0(\lambda_i^\alpha) - \tilde{v}_i^1(\lambda_i) + \underline{v}_i^1(\lambda_i^\alpha) \geq \epsilon, \tag{5.29}$$

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 $C = \underline{v}_i^0(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha)$. 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:

$$\Delta_i^F = \max_{\lambda_i \in \mathcal{F}_i(x^0)} [\tilde{v}_i^0(\lambda_i) - p^\phi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha))] \tag{5.30}$$

$$\text{s.t.} \quad (\tilde{v}_i^0(\lambda_i) - p^\phi(\lambda_i) - (\underline{v}_i^0(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha)))$$

$$- (\underline{v}_i^1(\lambda_i) - p^\phi(\lambda_i) - (\breve{v}_i^1(\lambda_i^\alpha) - p^\phi(\lambda_i^\alpha))) \geq \epsilon \tag{5.31}$$

$$= -C + \max_{\lambda_i \in \mathcal{F}_i(x^0)} \tilde{v}_i^0(\lambda_i) - p^\phi(\lambda_i) \tag{5.32}$$

$$\text{s.t.} \quad \tilde{v}_i^0(\lambda_i) - \underline{v}_i^0(\lambda_i^\alpha) - \underline{v}_i^1(\lambda_i) + \breve{v}_i^1(\lambda_i^\alpha) \geq \epsilon, \tag{5.33}$$

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 (5.31) 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 (5.33) 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* $\Delta_i^P \geq \Delta_i^F$.

5.3.4 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. (Detailed in Section 5.3.4.1.)

II: Fairness (FAIR) Second, we break ties to prefer prices that minimize the maximal deviation from provisional payments across all bidders. (Detailed in Section 5.3.4.1.)

III: Balance (BAL) Third, we break ties to prefer prices that minimize the maximal difference in price across all items. (Detailed in Section 5.3.4.1.)

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 5.5.

Example 5.9: *Consider the example in Figure 5.10 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*

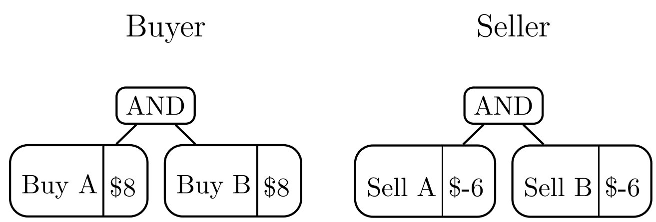


Figure 5.10: A simple example to illustrate pricing. ACC prices AB between \$12 and \$16, FAIR narrows this to \$14 and BAL requires $A = \$7$, $B = \$7$

values is for the trade to complete. ACC requires $12 \leq \phi_A + \phi_B \leq 16$*, 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* $\phi_A + \phi_B = 14$*. Finally, BAL requires* $\phi_A = \phi_B = 7$.

5.3.4.1 The Three Pricing Stages

Leaving aside the idea of lexicographical minimization for a moment, we first define the objectives of each of the three stages.

Accuracy

Given provisional trade, λ^α, and valuation profile, v^α, we define maximally *accurate* approximate-EQ prices as those that solve the following LP:

$$\begin{aligned}
&\delta^*_{\text{acc}} = \min_{\phi, \delta_{\text{acc}}} \delta_{\text{acc}} \\
\text{s.t.}\quad & v_i^\alpha(\lambda'_i) - \sum_j \phi_j \lambda'_{ij} \leq v_i^\alpha(\lambda_i^\alpha) - \sum_j \phi_j \lambda^\alpha_{ij} + \delta_{\text{acc}}, \quad \forall i, \forall \lambda'_i \in \mathcal{F}_i(x^0) \qquad (5.34)\\
& \delta_{\text{acc}} \geq 0, \\
& \phi_j \geq 0, \quad \forall j \in G
\end{aligned}$$

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 $\theta_i^\phi(\lambda_i^*, \lambda_i^\alpha, v_i^\alpha)$, where $\lambda_i^* = \text{argmax}_{\lambda_i \in \mathcal{F}_i(x^0)}[v_i^\alpha(\lambda_i) - p^\phi(\lambda_i)]$. 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.

Fairness

Second, we break the remaining ties to prefer *fair* prices: choosing prices that minimize the worst-case error in utility for each bidder with respect to the utility that would be achieved given Threshold payments at provisional valuation profile v^α. The fairness tie-breaking method is formulated as the following LP:

$$\delta^*_{\text{fair}} = \min_{\phi, \delta_{\text{fair}}} \quad \delta_{\text{fair}} \qquad \text{[FAIR]}$$

$$\text{s.t.} \quad v_i^\alpha(\lambda_i') - \sum_j \phi_j \lambda_{ij}' \leq v_i^\alpha(\lambda_i^\alpha) - \sum_j \phi_j \lambda_{ij}^\alpha + \delta^*_{\text{acc}}, \quad \forall i, \forall \lambda_i' \in \mathcal{F}_i(x^0) \qquad (5.35)$$

$$\delta_{\text{fair}} \geq U_{\text{vcg},i} - \sum_j \phi_j \lambda_{ij}^\alpha, \quad \forall i \qquad (5.36)$$

$$\delta_{\text{fair}} \geq \sum_j \phi_j \lambda_{ij}^\alpha - U_{\text{vcg},i}, \quad \forall i \qquad (5.37)$$

$$\delta_{\text{fair}} \geq 0, \quad \phi_j \geq 0, \quad \forall j \in G,$$

where $U_{\text{vcg},i}$ denotes the utility to bidder i in the VCG outcome given valuations v_i^α and trade λ_i^α. The objective here is the same as in the Threshold payment rule: minimize the maximal error between bidder payoff (at v^α) for the provisional trade and the VCG payoff (at v^α). For computational speed, rather than compute the Threshold payments separately, and then define prices to best approximate these payments, we do both in one step by using this formulation.

Balance

Third, we break the remaining ties to prefer *balanced* prices: choosing prices that minimize the maximal price difference across all items. For this, we solve a lexicographic sequence of LPs, alternating between minimization and maximization of all prices not yet pinned down. For instance, the minimization LP is formulated as follows (the maximization LP is an exact mirror):

$$\min_{\phi,\delta_{\text{bal}}} \quad \delta_{\text{bal}} \tag{BAL}$$

$$\text{s.t.} \quad v_i^\alpha(\lambda_i') - \sum_j \phi_j \lambda_{ij}' \le v_i^\alpha(\lambda_i^\alpha) - \sum_j \phi_j \lambda_{ij}^\alpha + \delta^*_{\text{acc}}, \forall i, \forall \lambda' \in \mathcal{F}(x^0) \tag{5.38}$$

$$\delta^*_{\text{fair}} \ge U_{\text{vcg},i} - \sum_j \phi_j \lambda_{ij}^\alpha, \quad \forall i \tag{5.39}$$

$$\delta^*_{\text{fair}} \ge \sum_j \phi_j \lambda_{ij}^\alpha - U_{\text{vcg},i}, \quad \forall i \tag{5.40}$$

$$\delta_{\text{bal}} \ge \phi_j, \quad \forall j \tag{5.41}$$

$$\phi_j \le \min\max \delta^*_{\text{bal}} + \epsilon, \quad \forall j \in G \tag{5.42}$$

$$\phi_j \ge \max\min \delta^*_{\text{bal}}, \quad \forall j \tag{5.43}$$

$$\delta_{\text{bal}} \ge 0, \phi_j \ge 0, \quad \forall j \in G,$$

Here, $\min\max \delta^*_{\text{bal}}$ and $\max\min \delta^*_{\text{bal}}$ represent the minimum δ_{bal} value and the maximum δ_{bal} value from all previous BAL maximization and minimization MIPs respectively. These two constraints serve to bind the prices into a progressively narrowing range that tends towards the center.

5.3.4.2 Lexicographical Refinement

For all three pricing stages, we also perform lexicographical refinement (with respect to bidders in ACC and FAIR, and with respect to goods in BAL). In addition to further improving the quality of the prices this also ensures uniqueness. For instance, in ACC we successively minimize the maximal error across all bidders. Given an initial solution we first "pin down" the error on all bidders for whom constraint (5.34) is binding. For such a bidder i, the constraint is replaced with

$$v_i^\alpha(\lambda_i') - \sum_j \phi_j \lambda_{ij}' \leq v_i^\alpha(\lambda_i^\alpha) - \sum_j \phi_j \lambda_{ij}^\alpha + \delta_{\mathrm{acc},i}^*, \quad \forall \lambda_i' \in \mathcal{F}(x^0), \tag{5.44}$$

and ACC is resolved with variable δ_{acc} included in Constraints (5.45) for bidders not yet pinned-down, so that further progress is made in lexicographically minimizing the maximal error. Eventually, the sequence of ACC LPs terminate with a best-case error $\delta_{\mathrm{acc},i}^*$ defined for each bidder. ACC then passes this vector $\delta_{\mathrm{acc}}^* = (\delta_{\mathrm{acc},1}^*, \ldots, \delta_{\mathrm{acc},n}^*)$, to FAIR. Constraints (5.35) in FAIR are replaced with an individualized constraint, reflecting $\delta_{\mathrm{acc},i}^*$, for each bidder. A similar lexicographical optimization process is then used for FAIR, with constraints (5.36) and (5.37) replaced with $\delta_{\mathrm{fair,i}}^* \geq U_{\mathrm{vcg},i} - \sum_j \phi_j \lambda_{ij}^\alpha$ and $\delta_{\mathrm{fair,i}}^* \leq \sum_j \phi_j \lambda_{ij}^\alpha - U_{\mathrm{vcg},i}$ as bidders i are "pinned down" and the payment error $\delta_{\mathrm{fair,i}}^*$ is lexicographically minimized for each bidder.

Upon termination of FAIR, the vector, $\delta_{\mathrm{acc}}^* = (\delta_{\mathrm{acc},1}^*, \ldots, \delta_{\mathrm{acc},n}^*)$, is passed together with $\delta_{\mathrm{fair}}^* = (\delta_{\mathrm{fair,1}}^*, \ldots, \delta_{\mathrm{fair,n}}^*)$ from FAIR into BAL. In addition to constraints (5.38), constraints (5.39) and (5.40) are also fixed in BAL, to enforce these $\delta_{\mathrm{fair,i}}^*$ values for each bidder. Finally, BAL proceeds lexicographically as well, but over all goods G. The result of this three-stage price generation algorithm is a set of unique linear prices that are not guaranteed to be accurate, fair or balanced — but they are "as close as possible" to having these properties given the limited power of linear prices.

5.3.4.3 Constraint Generation

Problems ACC, FAIR and BAL all have an exponential number of constraints because the price accuracy constraints (5.34) are defined for all trades $\lambda' \in \mathcal{F}(x^0)$ and all bidders i. It is unfeasible to even write this problem down. Rather than solve it explicitly we use *constraint generation* [e.g. Bertsimas and Tsitsiklis, 1997] and dynamically generate a sufficient subset of constraints. Constraint generation (CG) considers a relaxed program that contains only 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 to find a "violated constraint" in the full problem that is then introduced allowing the (now strengthened) relaxed program to be resolved.

We illustrate this process for ACC. Let $\mathbb{F}_i$ denote a manageable subset of all possible feasible trades to bidder i. Then, a relaxed version (called *ACC'*) is formulated by substituting constraints (5.34) with:

$$v_i^\alpha(\lambda_i') - \sum_j \phi_j \lambda_{ij}' \leq v_i^\alpha(\lambda_i^\alpha) - \sum_j \phi_j \lambda_{ij}^\alpha + \delta_{\text{acc}}, \ \forall i, \forall \lambda_i' \in \mathbb{F}_i, \tag{5.45}$$

where $\mathbb{F}_i$ is only a subset of trades that are feasible for bidder i given the other bids. We also maintain the trades in this set across pricing stages and across rounds of the exchange. Using this subset $\mathbb{F}$ instead of the full set of trades, $\mathcal{F}(x^0)$, has let us tractably specify the ACC MIP, but the MIP may not produce the correct solution unless we ensure that the necessary trades are included in $\mathbb{F}$.

How can we identify the important additional trades to include (i.e. those that correspond to the binding constraints of the original MIP)? Let ϕ^* denote the solution to relaxed problem, ACC'. We can solve n subproblems, one for each bidder, of the form:

$$\max_{\lambda_i'} \quad v_i^\alpha(\lambda_i') - \sum_j \phi_j^* \lambda_{ij}', \tag{R-WD(i)}$$

$$\text{s.t.} \quad \lambda_i' \in \mathcal{F}(x^0) \tag{5.46}$$

to check whether solution $(\phi^*, \delta^*_{\text{acc}})$ to ACC' is also a feasible solution to ACC. Problem R-WD(i), the "restricted winner-determination problem for bidder i", is a specialization of the WD problem, where the objective is modified to maximize the payoff of a single bidder at the given prices, rather than the total value across all bidders. It is solved as a MIP, by rewriting the objective in WD(T, x^0) as $\max\{v_i(\beta) \cdot sat_i(\beta) - \sum_j p^*_j \cdot \lambda_{ij}\}$ for bidder i. Thus, the structure of *TBBL* is exploited in generating new constraints, because this subproblem can be solved as a concise MIP. The trees submitted by bidders $i' \neq i$ are used to define the space of feasible trades.

Let $\breve{\lambda}_i$ denote a solution to R-WD(i). We check condition:

$$v_i^\alpha(\breve{\lambda}_i) - \sum_j \phi^*_j \breve{\lambda}_{ij} \leq v_i^\alpha(\lambda_i^\alpha) - \sum_j \phi^*_j \lambda^\alpha_{ij} + \delta^*_{\text{acc}}, \tag{5.47}$$

and if this condition holds for all bidders i, then solution $(\phi^*, \delta^*_{\text{acc}})$ is optimal for the full program ACC. Otherwise, trade $\breve{\lambda}_i$ is added to constraint set, $\mathbb{F}_i$, for the bidders i for which constraint (5.47) is violated and we re-solve ACC' with the new set of constraints. A similar constraint generation process is also used in the Fairness and Balance stages, as illustrated in Figure 5.11. Because constraint generation can only complete when it has established an adequate constraint sets $\mathbb{F}_i$, we endeavor to appropriately fill $\mathbb{F}_i$ as fast as possible as follows:

1. In the first round we employ a heuristic seed to establish an initial $\mathbb{F}_i$ set. (In addition to including both the provisional trade and the null trade).

2. In subsequent rounds we carry forward all trades already added to $\mathbb{F}_i$ in previous rounds of the exchange.

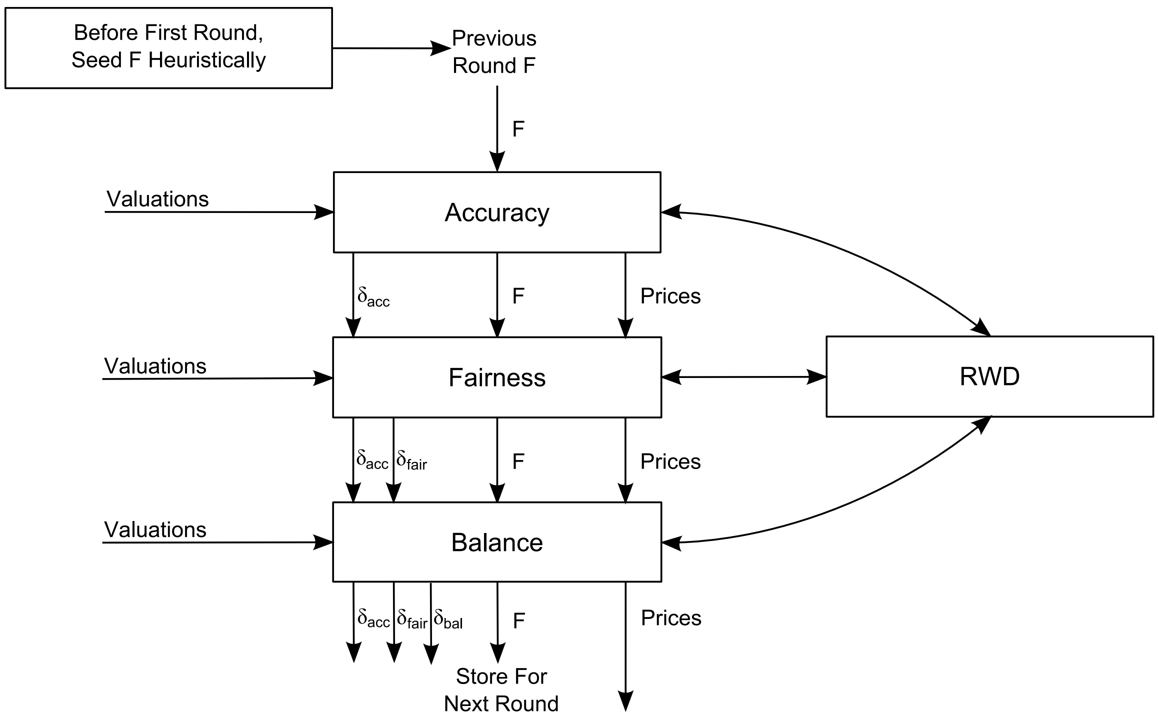


Figure 5.11: Using constraint generation to calculate prices. The constraint set (indicated F in the figure) is maintained across rounds and propagated across the stages. A restricted WD problem (R-WD) checks for violated constraints.

5.3.4.4 Heuristic Seeding of Constraint Sets

By maintaining the set of constraints generated in earlier rounds we place most of the computational cost on computing prices in the very first round of ICE, when the trade set, $\mathbb{F}$, is empty. To speed-up computation we use a heuristic method to seed $\mathbb{F}$. We heuristically guess what the prices should be, and then solve an R-WD(i) subproblem for each agent in order to back out appropriate trades to add to $\mathbb{F}$. We use a heuristic that sets the price on a good to its average contribution to the value of the provisional trade:

```
function midPointPrice(λ^α, T)
    for all goods g traded in λ^α do
        bCount := 0, sCount := 0, bVal := 0, sVal := 0
        for all bidders i ∈ I do
            B := buy nodes in T_i active in λ^α for good g
            S := sell nodes in T_i active in λ^α for good g
            bCount += |B|, sCount += |S|
            bVal += pathVal(B), sVal += pathVal(S)
        end for
        p_g := max(0, 0.5 · (bVal/bCount) + 0.5 · (sVal/sCount))
    end for
    for all goods g not traded in λ^α do
        bVal = max_{β∈Buys(g)} pathVal(β)/quantity(β)
        sVal = max_{β∈Sells(g)} pathVal(β)/quantity(β)
        p_g := 0.5 · bVal + 0.5 · sVal
    end for
end function
function pathVal(N)
    v := 0
    for all nodes n in N do
        for all ancestors a of n do
            v += v^α(a) / avgActiveDescendents(a)
        end for
    end for
    return v
end function
```

Here, avgActiveDescendents(β) denotes the average of the minimum number and the maximum number of descents of β that can be activated subject to the IC constraints.

5.3.4.5 Streamlining of Computation

Within a given pricing stage (i.e., ACC, FAIR or BAL) we must perform CG together with lexicographical refinement. We have streamlined this computation with two techniques:

Provisional Locking There are two types of mathematical programs we solve in pricing: LPs that solve for prices and IPs that generate additional constraints. The LPs can be solved significantly faster then the IPs, but several IPs can be run in parallel because they optimize over a single agent rather then globally. As a consequence, rather then performing our lexicographic ordering in each stage by locking one value at a time, we greedily run a sequence of LPs without CG, provisionally locking down at least $m_{\text{lock}} \geq 1$ values (e.g., δ^*_{acc} values in ACC, for a sequence of agents), before checking for violating constraints. The best choice for m_{lock} depends on the relative speed with which the LPs and IPs can be solved. In our experiments, we have found that choosing m_{lock} to be aggressive (i.e. close to 100% of the number of values), so that nearly all decisions are made before checking the constraints, works well.

Lazy Constraint Checks When performing CG, we choose not to check the validity of current prices for every bidder every time. Instead, once a bidder i has been found to pass, we optimistically assume that the error δ_i computed for that bidder remains valid, only running the expensive IP check for the bidders which continue to fail. Only once there are no such failing bidders, do we perform an additional check of *all* bidders to ensure that the prices have remained stable enough for the validity of the skipped bidders to have been maintained. If this check fails for any bidder, then CG must continue. Eventually, when all bidders pass simultaneously, we make the provisional locks permanent, and continue on to the next set of provisional locks in the lexicographic refinement. This process is illustrated for ACC in Figure 5.12. The other two stages follow similarly.

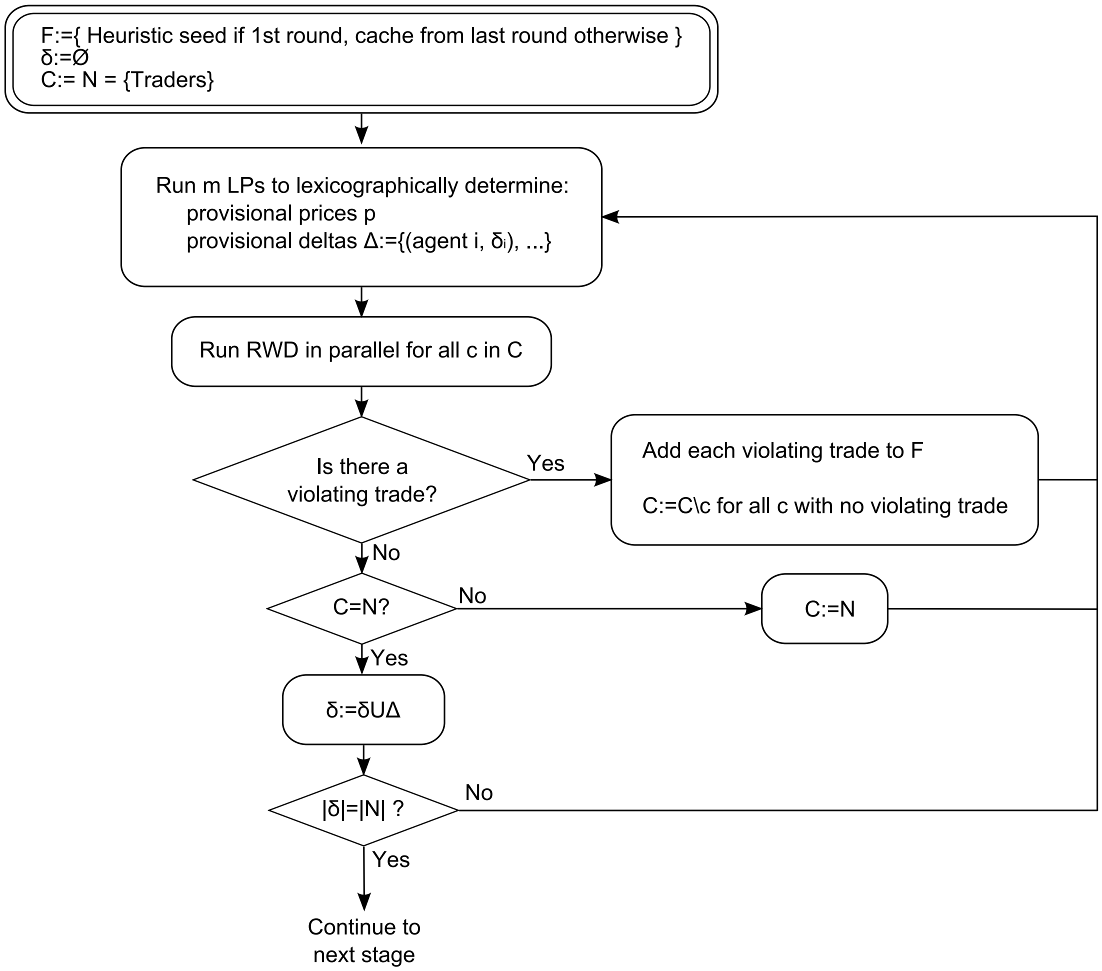


Figure 5.12: Streamlined computation for interleaving constraint generation with lexicographical optimization in the ACC pricing stage

Together these techniques can significantly improve the performance of the linear pricing engine, as illustrated in Figure 5.13.

5.3.5 Establishing Bounds on Efficiency

Consider some round t in ICE. The round starts with the announcement of prices—let us denote them ϕ^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*

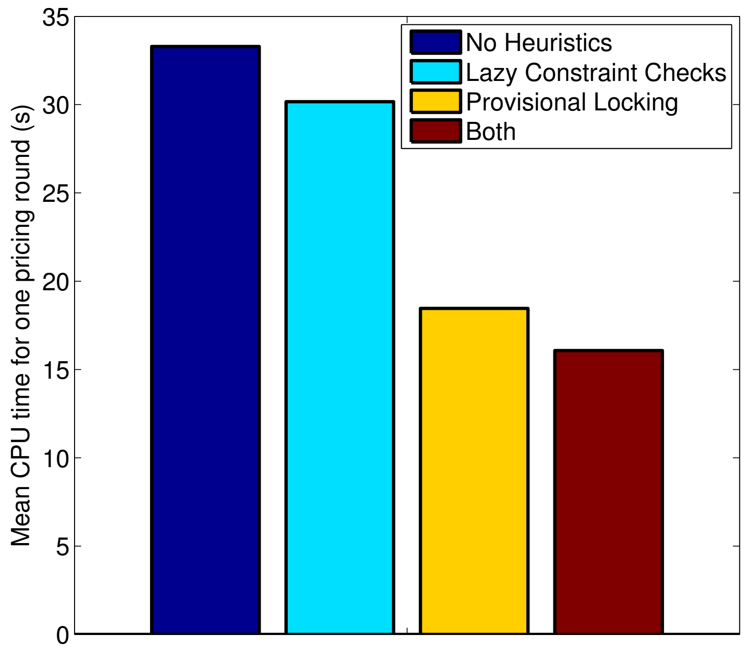


Figure 5.13: Effect of lacy constraint checks and provisional locking on the speed of pricing rounds.

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 provides a concluding 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 $\underline{\lambda} \in \mathrm{argmax}_{\lambda \in \mathcal{F}(x^0)} \sum_i \underline{v}_i(\lambda_i)$ denote the trade that is optimal given the lower bound valuations. As explained in Section 5.3.1, ICE is parameterized by a 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 $\mathrm{EFF}(\underline{\lambda})$ 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.

5.3.5.1 A Price-Based Proof of Efficiency

We have already seen in Section 5.3.3.1 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 ϕ^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 2.8.

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,

$$\text{EFF}(\underline{\lambda}) = \frac{\sum_i v_i(\underline{\lambda}_i)}{\sum_i v_i(\lambda^*)} \geq 1 - \frac{2\min(M, \frac{n}{2})}{\sum_i v_i(\lambda_i^*)}\delta \geq 1 - \frac{2\min(M, \frac{n}{2})}{\max_{\lambda \in \mathcal{F}(x^0)} \sum_i \overline{v}_i(\lambda)}\delta, \tag{5.48}$$

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 $\omega^{\text{price}} := 0$.

5.3.5.2 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. To establish this bound we need the definition of *perturbed valuations* that was introduced formally in Section 5.3.3.1. Informally, a *perturbed valuation* for a given *TBBL* tree is defined to be the lower bound value at all the nodes satisfied by a specific trade, and the upper bounds everywhere else. Using this concept we can establish the following efficiency bound directly from the valuation bounds in the *TBBL* trees:

$$\text{EFF}(\underline{\lambda}) = \frac{v(\underline{\lambda})}{v(\lambda^*)} \geq \min_{v' \in T, \lambda' \in \mathcal{F}(x^0)} \left[\frac{v'(\underline{\lambda})}{v'(\lambda')}\right] = \min_{\lambda' \in \mathcal{F}(x^0)} \left[\frac{\tilde{v}(\underline{\lambda})}{\tilde{v}(\lambda')}\right] = \frac{\underline{v}(\underline{\lambda})}{\tilde{v}(\tilde{\lambda})} \equiv \omega^{\text{direct}} \tag{5.49}$$

where 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 arrived at through perturbed valuation $\tilde{v}$. For the final equality, $\tilde{v}(\underline{\lambda}) = \underline{v}(\underline{\lambda})$ by definition,

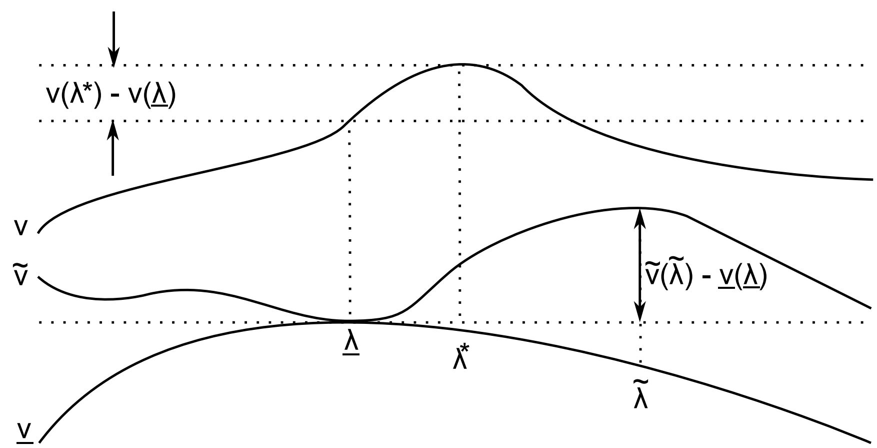


Figure 5.14: Determining an efficiency bound based on lower and upper valuations.

and the optimal trade λ' is that which maximizes the value of the denominator, i.e., trade $\tilde{\lambda}$. Figure 5.14 schematically illustrates the various trades and values used in this bound, and in particular provides some graphical intuition for why $\tilde{v}(\tilde{\lambda}) - \underline{v}(\underline{\lambda}) \geq \tilde{v}(\lambda^*) - \underline{v}(\underline{\lambda}) = \max_{v' \in T}[v'(\lambda^*) - v'(\underline{\lambda})] \geq v(\lambda^*) - v(\underline{\lambda})$.

5.3.5.3 Combining the Bounds 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 5.10, we immediately get our main result.

Theorem 5.11: *When ICE incorporates δ-MRPAR and ϵ-DIAR and when all bidders are straightforward, then the exchange terminates with a trade that is* *within the target approximation error Δ^*, for any $\Delta^* \geq 0$ as $\epsilon \rightarrow 0$.*[11]

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 5.5. For the ϵ parameter in ϵ-DIAR, we find that a simple rule:

$$\epsilon := \frac{1}{2n} \sum_i \sum_{\beta \in T_i} \frac{\overline{v}_{i \in N}(\beta) - \underline{v}_i(\beta)}{|T_i|}, \tag{5.50}$$

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.[12]

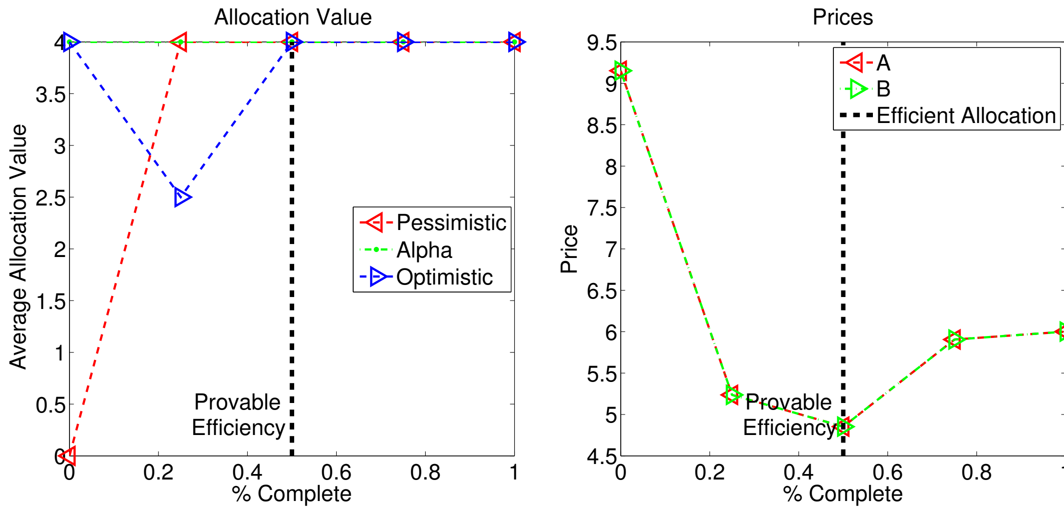
[11] In practice, we choose ϵ endogenously, reducing it only when progress is not being made and efficiency can not yet be proven. Theoretically ϵ may need to approach zero to reach efficiency. But in all observed instances it remains a reasonably large positive constant.

[12] 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.

5.4 Illustrative Examples

In this section we illustrate the behavior of the exchange on two simple examples. These examples are furnished 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. We defer a detailed discussion of the operation of these agents until Section 5.5.

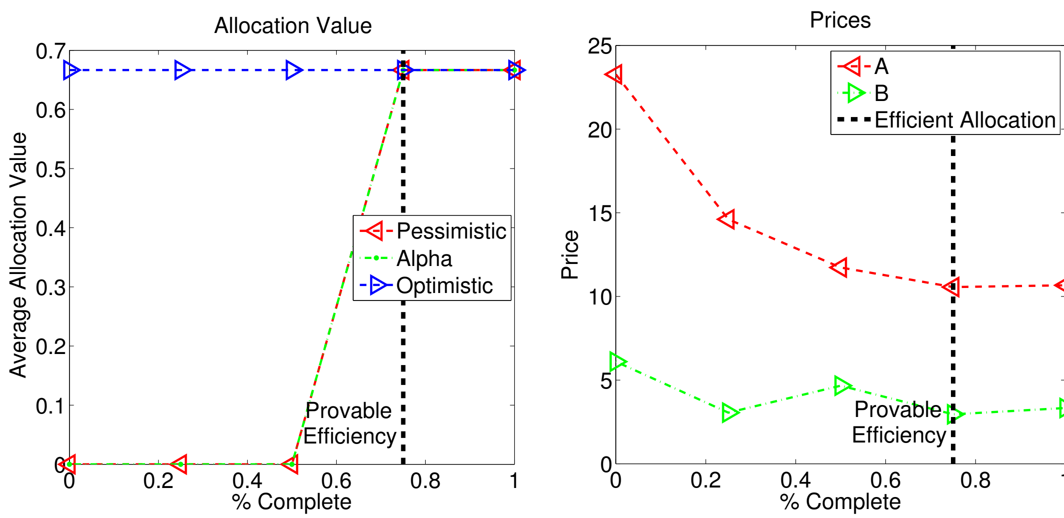
In this section, and also in presenting our main experimental results, we do *not* move to a last-and-final round. Rather, the bidding agents are programmed to continue to improve their bids past the round at which efficiency is already proved (and when a last-and-final round would ordinarily be declared), 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 5.15: AgentA: **A** \$8, AgentB: **B** \$8, AgentAB: **A** AND **B** \$10. In these graphs the horizontal axis stipulates the round of the mechanism, normalized so that termination occurs at 100%. For experimental purposes, we hold ICE open until the payments charged match those that would occur at truth. In actual use, the exchange would likely be closed at the point where the efficient trade can be determined, here indicated by the vertical dashed line.

Example 5.10: *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 5.15a 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 5.15b 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 5.16: Seller **A** -\$10, Swapper: swap **B** for **A** \$8, Buyer **B** \$4

Example 5.11: *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 5.16a, for a trade to be found in the pessimistic valuation. Revelation drives progress towards a completed trade, and as we can see in Figure 5.16b, 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****.*

5.5 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.

5.5.1 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 all the 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 5.12: *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 5.13: *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 ϕ. Then, there exists a trade $\lambda_i^* \in \mathcal{F}_i(x^0)$ such that $\theta_i^\phi(\lambda_i^*, \breve{\lambda}_i, v_i) > 0$. But, this is a contradiction with MRPAR, since $\theta_i^\phi(\breve{\lambda}_i, \lambda_i', v_i') \geq 0$ for all $v_i' \in T_i$ and all $\lambda_i' \in \mathcal{F}_i(x^0)$, including $v_i' = v_i$ and $\lambda_i' = \lambda_i^*$. $\square$

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

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

5.5.2 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 5.1.[13] The prototype is modular so that researchers may easily replace components for experimentation. 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 arise in 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 chapter 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[14] (we currently support

[13]Code size is measured in physical source line of code (SLOC).

[14]`http://www.eecs.harvard.edu/econcs/jopt`

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 5.1: Exchange components and code breakdown

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.

5.5.3 Experimental Setup

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 5.3.3.3. 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. In 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 IV 3.2GHz with 8GB of memory and CPLEX 10.1. All results are averaged over 10 trials.

5.5.4 Problem Instance Generator

Because we need CE and not CA instances, and because we need these instances specified in our concise *TBBL* language, we cannot use existing problem generators such as CATS [Leyton-Brown et al., 2000]; instead we must create our own. 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*.[15] 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.[16] 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.[17]

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

[16] 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.

[17] In our experiments, we vary $2 \leq |G| \leq 128$, $1 \leq d \leq 128$, $2 \leq |N| \leq 20$, $2 \leq \mathit{outDegreeLow} \leq 8$, $2 \leq \mathit{outDegreeHigh} \leq 8$, $2 \leq \mathit{depthLow} \leq 6$, $2 \leq \mathit{depthMid} \leq 6$, $2 \leq \mathit{depthHigh} \leq 8$, 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 $[-25, 25]$.

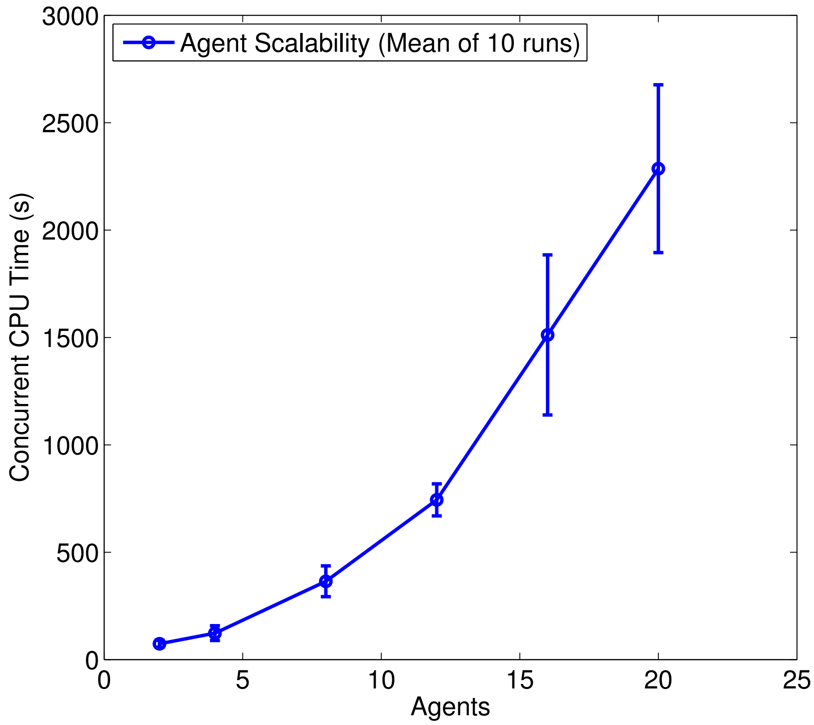


Figure 5.17: Effect of the number of bidders on the run-time of ICE

5.5.5 Experimental Results: Scalability

The first set of results that we present focuses on the computational properties of ICE. Figure 5.17 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 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 5.18 we can see the effect of varying the number of types of goods

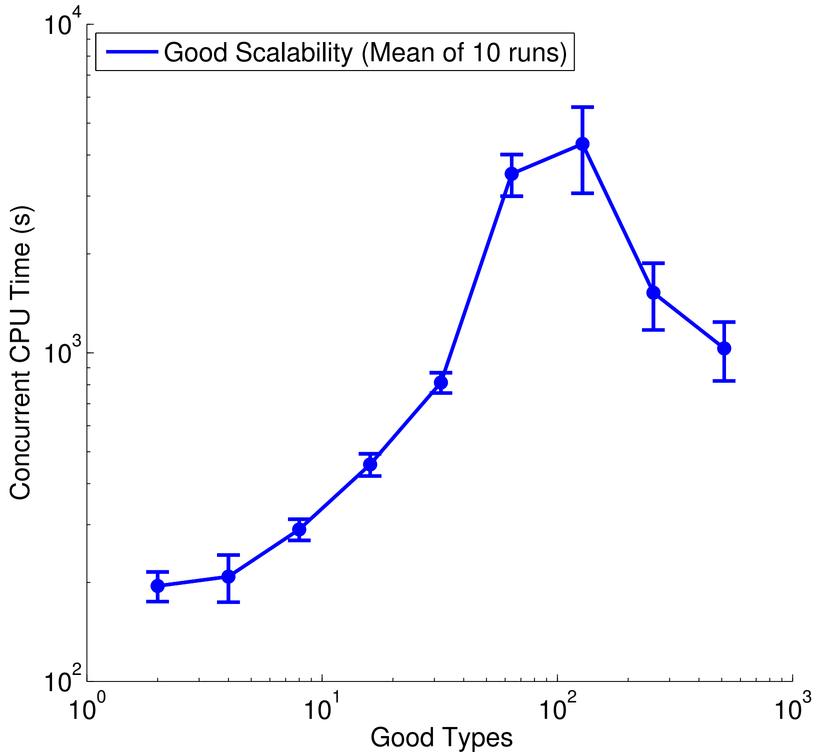


Figure 5.18: Effect of the number of good types on the run-time of ICE

(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 5.19). The average price provides a good proxy for the competitiveness of the market. Adding 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 5.20 and 5.21 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

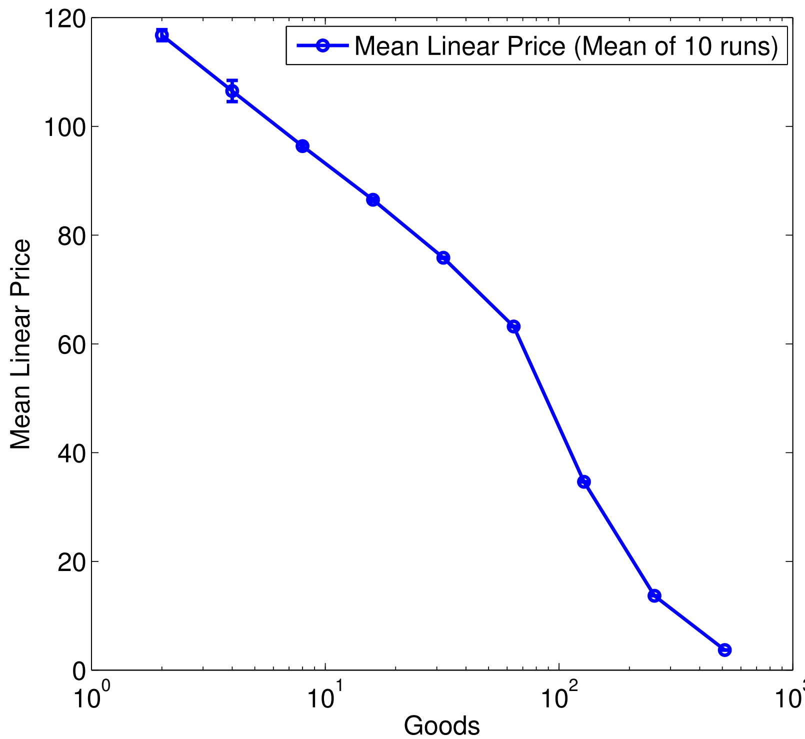


Figure 5.19: Effect of the number of goods on the average item price upon termination of ICE.

were being traded by 10 bidders. In Figure 5.20 we vary the number of children of any given node while in Figure 5.21 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.

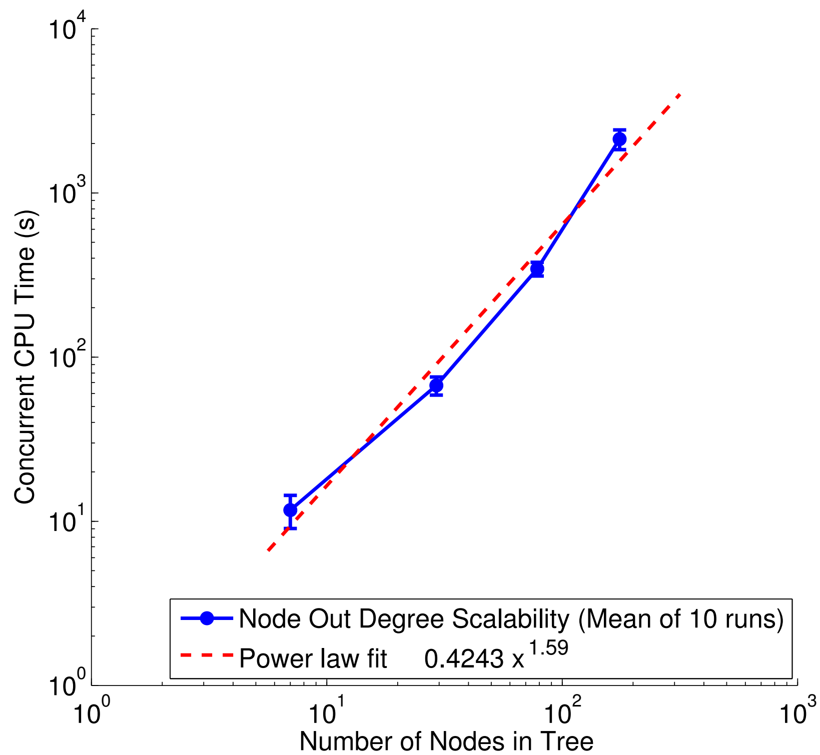


Figure 5.20: Effect of bid-tree size on run-time of ICE: Varying the node-out degree.

5.5.6 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 5.22 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

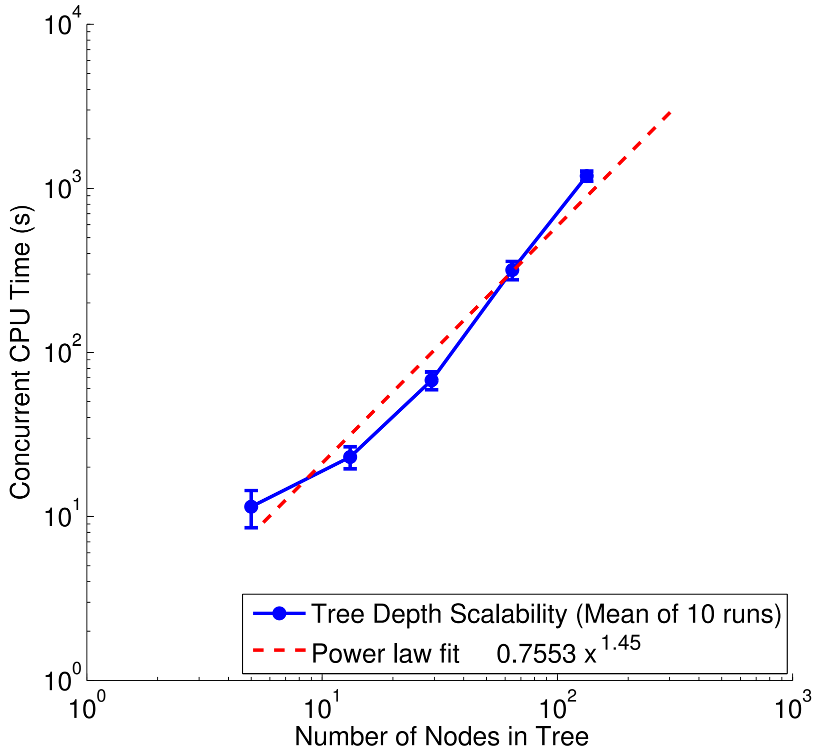


Figure 5.21: Effect of bid-tree size on run-time of ICE: Varying the tree depth.

which enables results to be aggregated across multiple instances, each of which can have a different number of total rounds.[18] 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 5.22, we see that the exchange quickly converges to highly efficient trades, taking an average of 6.8 rounds to achieve efficiency. In general, the opti-

[18]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.

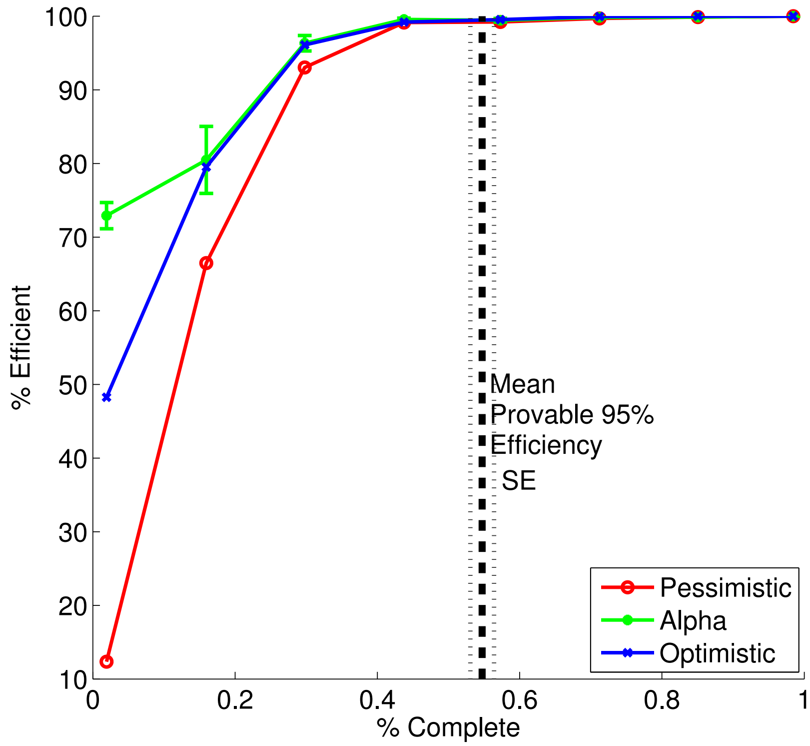


Figure 5.22: Efficiency of the optimistic, provisional, and pessimistic trades across rounds.

mistic 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 5.23 shows the average amount of revelation caused by MRPAR and DIAR

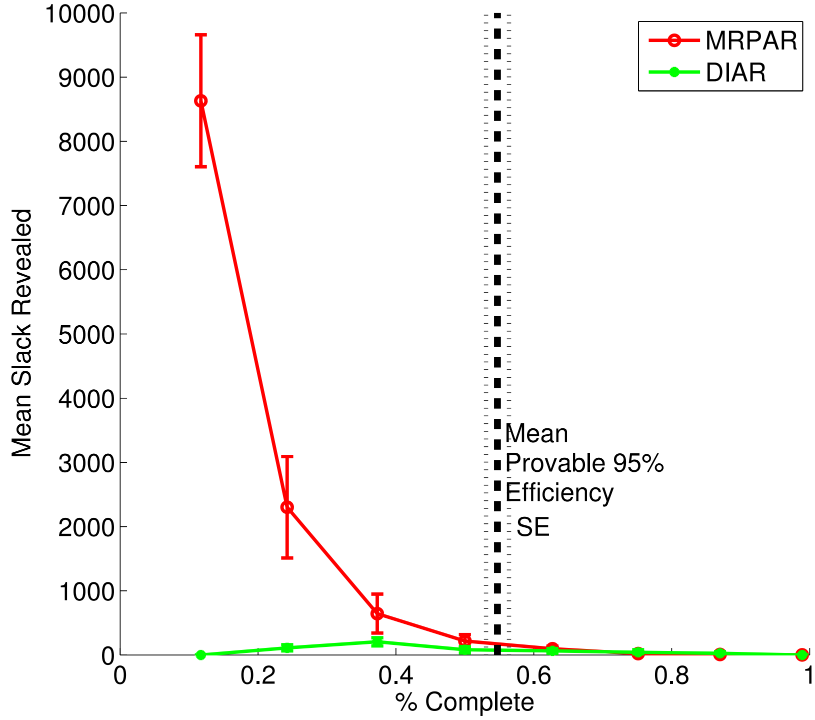


Figure 5.23: Average reduction in value uncertainty due to each rule.

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 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

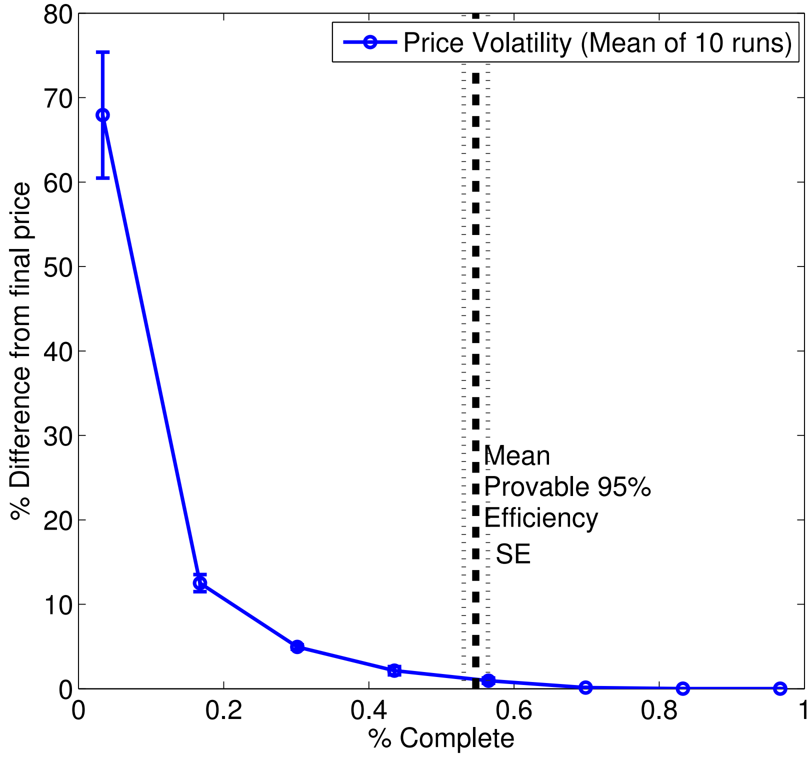


Figure 5.24: Price trajectory: Closeness of prices in each round to the final prices

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 5.24 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 5.25 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

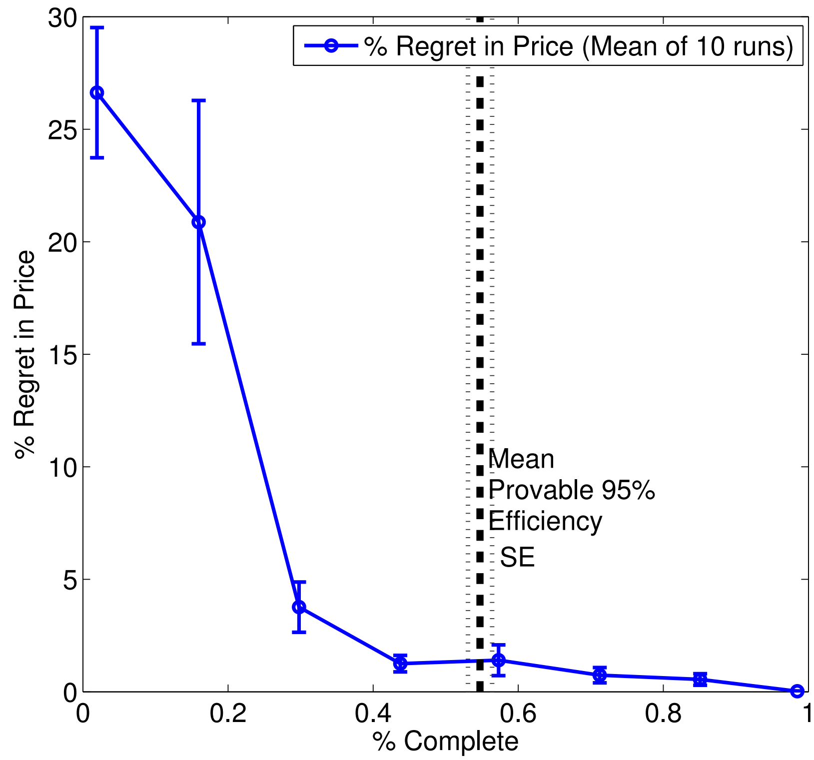


Figure 5.25: Regret in best-response by bidders due to price inaccuracy relative to final prices.

lost payoff at those final prices. Define the regret to bidder i for his best response $\lambda_i' = \text{argmax}_{\lambda_i \in \mathcal{F}_i(x^0)}[v_i(\lambda_i) - p^{\hat{\phi}}(\lambda_i)]$, to prices $\hat{\phi}$, given that the final prices are ϕ, as:

$$\text{Regret}_i(\lambda_i', \phi) = \left(1 - \frac{v_i(\lambda_i') - p^{\phi}(\lambda_i')}{\max\limits_{\lambda_i \in \mathcal{F}_i(x^0)} v_i(\lambda_i) - p^{\phi}(\lambda_i)}\right) \times 100\%. \tag{5.51}$$

As the payoff from trade λ_i', when evaluated at prices ϕ, approaches that from the best-response trade at prices ϕ, then $\text{Regret}_i(\lambda_i', \phi)$ approaches 0%. Figure 5.25 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

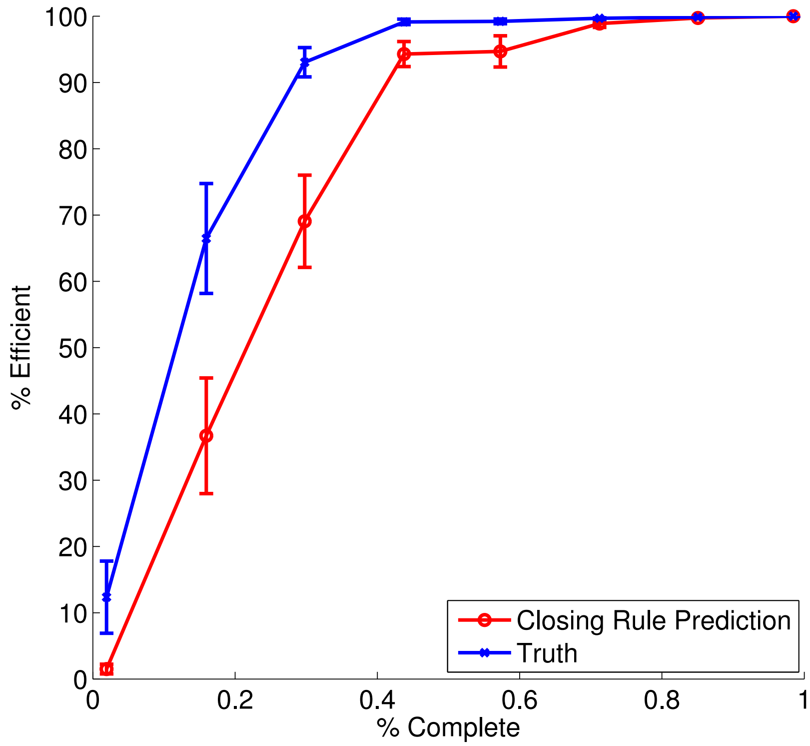


Figure 5.26: Comparison between the actual efficiency of the pessimistic trade and the ω^{direct} bound.

proceed.

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 5.26 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.

5.6 Related Work

Many ascending-price one-sided CAs are known in the literature [Ausubel and Milgrom, 2002; de Vries, Schummer, and Vohra, 2007; Mishra and Parkes, 2007; Parkes and Ungar, 2000a; Wurman and Wellman, 2000]. Direct elicitation approaches, in which bidders respond to explicit queries about their valuations, have also been proposed for one-sided CAs [Conen and Sandholm, 2001; Hudson and Sandholm, 2004; Lahaie and Parkes, 2004; Lahaie, Constantin, and Parkes, 2005; Sandholm and Boutilier, 2006]. Of particular relevance here are the ascending CAs that are designed to work with simple prices on items [Dunford, Hoffman, Menon, Sultana, and Wilson, 2003; Kwasnica, Ledyard, Porter, and DeMartini, 2005]. In computing (approximately competitive) linear prices, we generalize and extend these methods. Building on the work of Rassenti et al. [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 and 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 [Ausubel and Milgrom, 2002; Parkes and Ungar, 2000b]. 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 [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 and Satterthwaite, 1983]. Given this, Parkes et al. [2001a] study sealed-bid combinatorial exchanges and introduced the Threshold payment rule; 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, 1992b] and also for slightly more expressive domains [Babaioff and Walsh, 2005; Chu and Shen, 2008]. 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 and 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 [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 and 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 et al. [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 [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.

5.7 Discussion

In the work described in this chapter 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 can be no economy where there is no efficiency.

– Benjamin Disraeli
UK Prime Minister, 1868–1880

6

An Expressive Power-Based Market for Data Center Resources

6.1 Introduction

In 2006, US data centers used about 61 billion kWh; that is, 1.5% of the 4 trillion kWh consumed in total. This is the amount of energy used by 5.8 million average US households (5% of all households). Producing this power resulted in 37 million metric tons of CO_2, or 0.6% of the 5.9 billion metric tons released from all sources. This is roughly 16% of the total produced by the burning of jet fuel and more than that used to power TVs. This electricity cost US $4.5 billion and required a peak load capacity of about 7GW, more than double the level of consumption in 2000. Peak load capacity is expected to have doubled again by 2011 to 12GW, requiring the construction of 10 additional 500MW power plants [DOE; EIA; EPA]. Given the rapid, unabated rise in electrical power consumption and the associated financial and environmental costs, data center operators realize that the established practice of running large numbers of significantly under-utilized servers is no longer acceptable, and are eager for energy-saving solutions.

Market paradigms have often been proposed as useful ones paradigm for allocating limited computational resources and satisfying multi-criteria objectives. The earliest work on such markets was for time sharing on the PDP-11 in the 1960s by Sutherland [1968]. In the intervening years there have been proposals to use such methods in high performance computing, grid computing, as well as in data centers. However, existing proposals have deficiencies that can render them impractical for modern data centers. We propose a general method for overcoming these concerns, and illustrate this method's applicability to one specific environment. We offer a:

Realistic model of resources: We support a finer granularity of computational entity (e.g. core vs. server, which is especially important as multi-core machines become the norm) and finer control over power state of machines (e.g. Dynamic Voltage and Frequency Scaling (DVFS), not just on/off). We also handle heterogeneous applications running on heterogeneous classes of servers.

Realistic representation of goals: We use a less restricted form of utility function that supports distributional (and percentile) considerations, not just means. These functions are derived from standard long-term Service Level Agreements (SLAs) that are programmatically interpreted as short-term utility functions in a dynamic environment.

Principled optimization: We use Mixed Integer Programming, and, unlike previous systems, we do not rely on heuristic solvers but instead present a carefully formulated MIP model that can scale to large problem sizes.[1]

We show that a market-based approach provides a natural, feasible, and advantageous framework for representing the milieu of physical and computational resources, and the applications that consume these resources, in modern-day data centers. Experimental results indicate that our system can robustly and scalably improve net profits of our data center prototype by up to 137%. For large instances of 1000 machines and 10 applications (each associated with a separate customer), we achieve an average solve time of 5.16 minutes when limiting MIP solve time to an absolute maximum of 10 minutes (an approximation that imposes a tiny efficiency loss on those instances that do timeout). Thus a single machine is capable of optimizing the usage on 1000 others, giving us a very acceptable 0.1% overhead factor.

This domain is a compelling place to apply the techniques that we have been studying in the preceding chapters of this thesis. Data centers present a complex combinatorial allocation problem to solve, with multiple self-interested participants whose conflicting goals need to be reconciled. While they do not constitute the focus of the particular work presented here, the incentive effects discussed in Chapters 3 and 4 do apply here. But perhaps even more importantly, the design principals and techniques that we presented in the *general* ICE mechanism in the last chapter, we can now simplify and reify in order to construct the *specific* mechanism proposed here.

[1] We leverage recent advances in MIP solving that enable complex combinatorial markets to be solved quickly in practice, even though they address NP-hard problems [Sandholm, 2007].

6.2 A Market Model of Data Center Allocation

Typical data centers have hundreds to thousands of servers, many of which will share the same hardware and software configuration. We call such equivalence classes *'machine groups'* and assume that this partitioning is performed by a separate offline process. The owner of a data center typically contracts (either internally or externally) to provide these resources to a set of applications (each associated with a customer), each with time-varying load and utility and a range of resource requirements and importance.

In present use, each application is associated with an SLA that is negotiated between customer and data-center provider. Such an agreement specifies a price, a performance objective (e.g. a cap on the 95^{th} percentile of response time), and a penalty for failing to meet the objective. The SLA is useful for assigning a relative importance to the applications, but despite its quantitative feel it is generally used at present in only a qualitative way, as a guideline for personnel when manually configuring data-center operations. Yet, SLAs suggest a direction towards application utility functions that are highly relevant to obtaining reasonable performance in a power-constrained environment [Kephart and Das, 2007; Steinder, Whalley, Hanson, and Kephart, 2008]. In the present work, we introduce a system that adapts SLAs for the purpose of utility-based optimization of resource configurations.

When allocating resources in the data center we seek to optimize the operator's business value for the data center: i.e., the revenue net of costs. This means assigning (portions of) the machines from discrete machine groups to the various applications as well as specifying the power for each machine, and thus restraining overall consump-

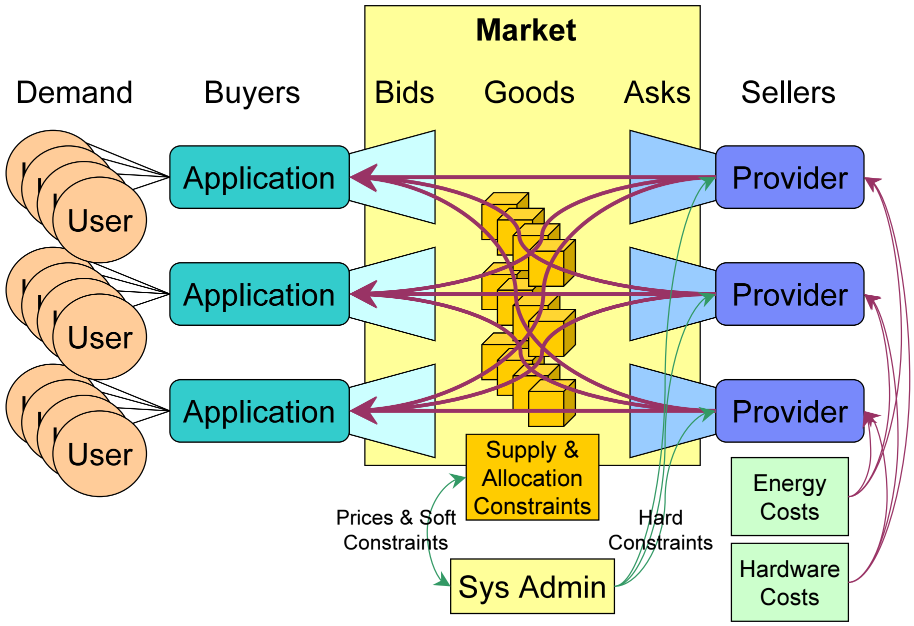


Figure 6.1: The data center market model

tion. For this, we use a sophisticated model of the power-saving modes available to modern servers and assume access to monitors of both power consumption and application demand.

Our market allocates goods (cores of machines from the various machine groups) to applications, as illustrated in Figure 6.1. The market is repeatedly cleared over brief periods, by using predictions about future supply and demand to translate applications' long-term SLAs into short-term bids.

The winner-determination problem for this market requires optimization, and is potentially solved in many different ways. We choose to formulate it as a Mixed Integer Program and solve it via ILog CPLEX 11.1. The form of the buyer's (or customer's) value model and seller cost model have been chosen to ease formulation of the problem as a MIP, as sketched below.[2]

[2] The size of the formulation will grow linearly with the number of applications and machine

For each period, we use a myopic net revenue maximization objective:

$$\text{argmax} \quad \sum_{a \in A} V_a - \kappa E^{\text{TOTAL}} - H^{\text{TOTAL}},$$

where V_a is the value of the chosen allocation of machines to application (associated with a particular buyer) $a \in A$, κ is the dollar cost of a kW-hour of energy, E^{TOTAL} is the total energy used to establish and maintain the chosen allocation for the current period, and H^{TOTAL} is the dollar cost for the hardware. The objective is thus quite straightforward–the complexity comes from the constraints. We begin by defining the buyer value, V_a, i.e. the value associated with application a of some buyer.

6.2.1 Buyer Valuation Model

Today, the contracts signed for data center provisioning are typically in terms of SLAs. We model a per-application SLA contract as a piecewise linear function for the value of receiving a given response time at a given demand percentile. Figure 6.2 shows an example of an SLA value curve of this form for two different applications A and B. A bidding proxy represents each application within the market, and takes such an SLA and combines it with supply and demand prediction and an application performance model, to represent the SLA as a short-term bid; i.e. a bid that is appropriate for the period of time for which the market is cleared.

The bidding proxy needs a model of how a given supply of machines (and thus transaction capacity) and application demand for the next planning episode will trans-

groups. However, it will grow with the square of the number of power modes (since it encodes a transition matrix from one mode to the next). Fortunately, polynomial growth in the size of the MIP need not imply exponential growth in practical computation time, and we examine scalability in Section 6.3.1.

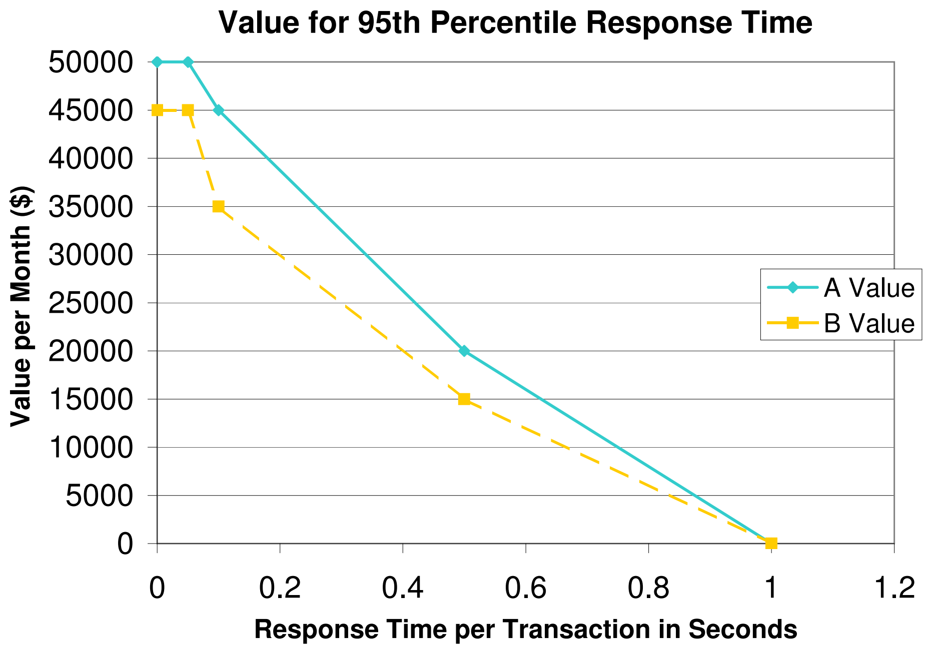


Figure 6.2: SLAs as provided by two different applications

late to the long-term response time distribution (and in turn to, e.g., its 95$^{\text{th}}$ percentile), and thus to the value curve associated with an SLA. Here, as a very simple example, we consider that transaction processing is described as an M/M/1 queue (exponential inter-arrival and service times). In this case, the response time distribution is exponential with mean response time $1/(\mu - \lambda)$, where μ is the supply and λ is the demand, both in transactions per unit time. The fraction of response times above a percentile P is given by the exponential quartile function: $-\frac{\ln(1-P)}{(\mu-\lambda)}$. The proxy composes the customer's SLA (Figure 6.2) with this response time model, resulting in a value function over both supply and demand at, e.g., the 95$^{\text{th}}$ percentile (Figure 6.3).[3]

Next the bidding proxy needs a predictive model of application demand over the

[3]The value is also scaled by the demand relative to the mean, aligning the one-period bid with the long-term (SLA) statistics.

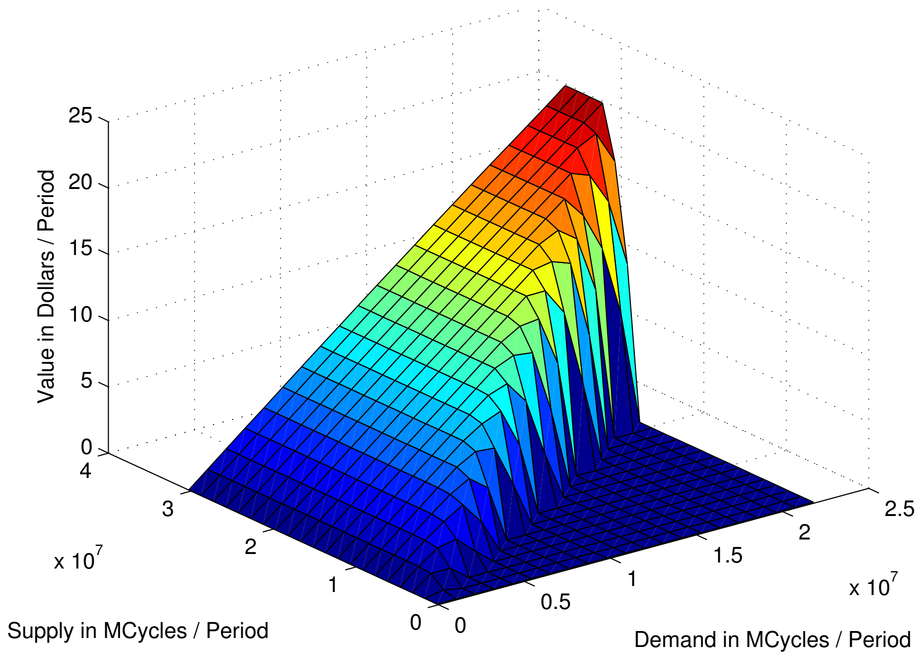


Figure 6.3: Application value by supply and demand

next period. We have found it sufficient to simply use statistics gathered over a small window of previous periods to provide a Gaussian model of the distribution of possible demand in the next period via a Maximum Likelihood Estimation (MLE) prediction of mean and standard deviation. The bidding proxy draws equal weighted samples from this Gaussian demand prediction model and takes a slice from the value model (Figure 6.3) for each. Then, these slices are averaged to produce a single supply-value curve, under our demand model. By taking a piecewise linear approximation of this curve (obtained by chaining the control points of the originally supplied response-time/value curve through these transformations) we arrive at the utility curve provided to the market in a given period as a bid, an example of which is shown in Figure 6.4.

If we apply this function to the cycles provided by a potential allocation, then we have specified a utility function as needed by the winner determination algorithm,

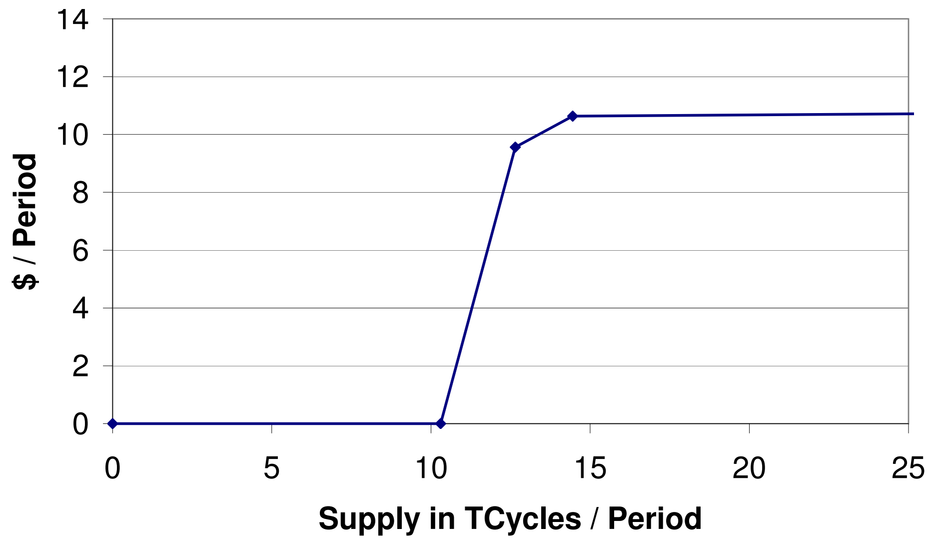


Figure 6.4: Expected short-term value for a single application

and with a significant dimensionality reduction:[4]

$$V_a = F_a(Q_a),$$

for application a, where F_a is this piecewise linear function and Q_a is the quantity of cycles provided to application a by the chosen allocation. To formulate this function, any standard MIP representation for a piecewise linear function can be used, which will induce auxiliary constraints and variables in order to account for the various segments of the function.

In turn, the total quantity of cycles provided to application a can in a period be defined as:

$$Q_a = \sum_{g \in G_a} \sum_{f \in M_g} \sum_{t \in M_g} \gamma_{g,t}(\tau - \delta_{g,f,t}) C^{\text{SOLD}}_{g,f,t,a} \; \forall \; a \in A$$

[4]Our queuing model permits a reduction from the $|G_a \times M_g \times M_g|$ variables to the single variable Q_a. More complex models may require additional dimensions, though in general a significant diminution should be possible.

where G_a is the set of machine groups that can support application a, M_g is the set of power modes available to machines in group g, $\gamma_{g,t}(\Delta)$ is the number of cycles provided by a machine from group g in mode t over a period of time Δ, τ is the amount of time in the current period, $\delta_{g,f,t}$ is the amount of time it takes to transition from mode f to mode t and each $C^{\text{SOLD}}_{g,f,t,a}$ variable defines a quantity of cores (i.e. goods) allocated from group g that were in mode f and are now in mode t (described in more detail below).

6.2.2 Defining The Goods in the Market

Within each machine group, we track only the number of cores in each power state. An allocation of some quantity of such cores is ultimately mapped into an assignment of cores on physical machines in post-processing.[5]This avoids the creation of immaterial distinctions that would only complicate winner determination. However, to properly encode the data-center cost model, described in the next section, we need a representation that captures power-state transitions enabling us to account for resultant changes in energy usage, cycle loss and increases in failure rate. Consequently, the $C^{\text{SOLD}}_{g,f,t,a}$ variables capture the number of cores in a given machine group starting in mode f in the last period, transitioning to (the possibly identical) mode t in the current period for a given application a.

Constraints are defined to ensure that an allocation of these goods will be physically implementable; e.g., on present-day platforms it is required that all cores on the

[5]We currently use a fast but potentially only approximate greedy assignment; however, more sophisticated methods could be used if the identities of machines in a group has importance.

same physical machine be at the same power level:

$$|\textsc{cores}_g| \sum_{f \in M_g} M^{\textsc{sold}}_{g,f,t} = \sum_{f \in M_g} \sum_{a \in A} C^{\textsc{sold}}_{g,f,t,a} + C^{\textsc{partUnsold}}_{g,t}$$

$$|\textsc{cores}_g| \sum_{f \in M_g} M^{\textsc{unsold}}_{g,f,t} = \sum_{f \in M_g} C^{\textsc{unsold}}_{g,f,t} - C^{\textsc{partUnsold}}_{g,t}$$

$$\forall\, t \in M_g \;\forall\, g \in G$$

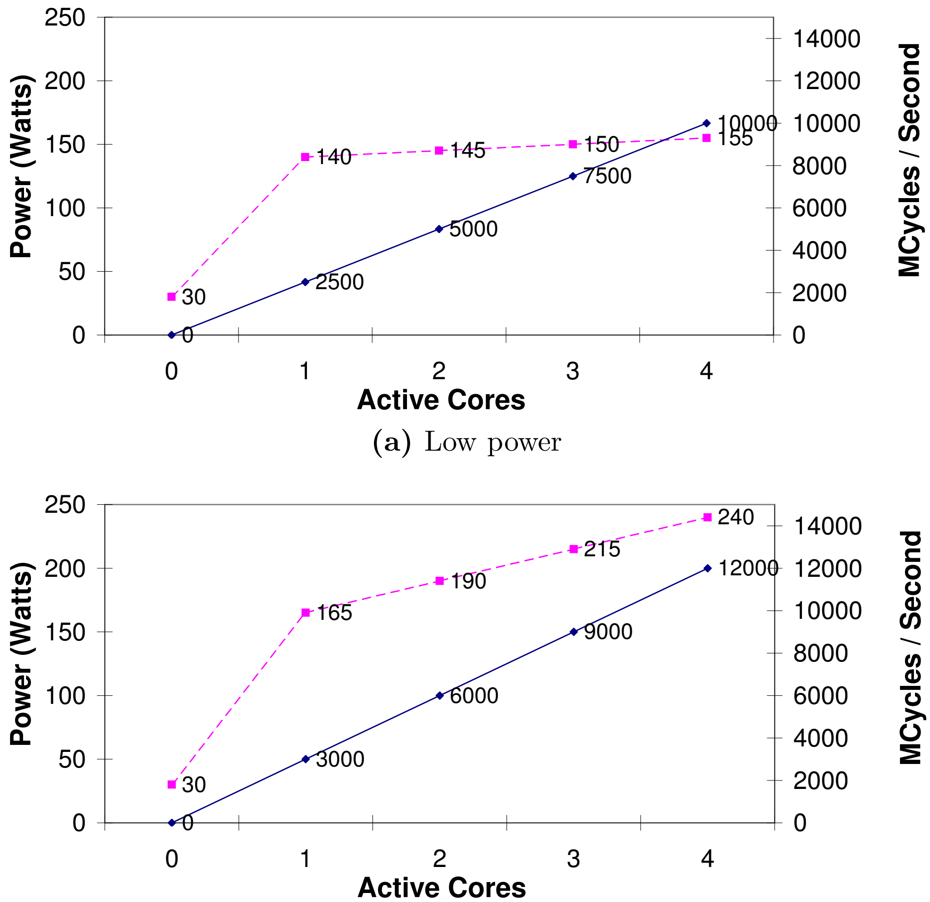
where $|\textsc{cores}_g|$ is the number of cores per machine in group g, $C^{\textsc{unsold}}_{g,f,t}$ are variables counting the unassigned cores, $M^{\textsc{sold}}_{g,f,t}$ and $M^{\textsc{unsold}}_{g,f,t}$ count sold and unsold machines respectively and $C^{\textsc{partUnsold}}_{g,t}$ count the unsold cores on partially sold machines. Additionally, we need to restrain available supply, through the machine counts:

$$|\textsc{machines}_g| = \sum_{f \in M_g} \sum_{t \in M_g} M^{\textsc{sold}}_{g,f,t} + M^{\textsc{unsold}}_{g,f,t} \;\forall\, g \in G$$

where $|\textsc{machines}_g|$ is the number of machines in group g.

6.2.3 Seller Cost Model

On the supply side of the market, we explicitly model both the hardware and energy costs of running the data center's machines in their various power states. Our model captures the power consumed and performance attained by each machine as a function of the number of active and inactive cores, as measured empirically on an IBM BladeCenter HS21 Server (Figures 6.5a and 6.5b). Modern *Dynamic Voltage and Frequency Scaling* (DVFS) enabled machines can have their most efficient state at less than full power: e.g. a maximum of 64 vs. 50 MCycles/Watt with 4 cores active (taking the ratio of the curves in each figure).



(a) Low power

(b) High power

Figure 6.5: Power and speed under low and high power

We define the energy requirements (i.e. power over the time period) of the active cores as follows (omitting that for idle hardware in the interest of space):

$$E^{\text{SOLD}} = \sum_{g \in G} \sum_{f \in M_g} \sum_{t \in M_g} \mathrm{E}^{\text{MULT}} (\mathrm{E}^{\text{TRANS}}_{g,f,t} + \mathrm{E}^{\text{BASE,ACTIVE}}_{g,\tau,t}) M^{\text{SOLD}}_{g,f,t}$$

$$+ \sum_{g \in G} \sum_{f \in M_g} \sum_{t \in M_g} \sum_{a \in A} \mathrm{E}^{\text{MULT}} \mathrm{E}^{\text{CORE,ACTIVE}}_{g,\tau,t} C^{\text{SOLD}}_{g,f,t,a}$$

where $\mathrm{E}^{\text{TRANS}}_{g,f,t}$ is the energy required to go from power-state f to t, $\mathrm{E}^{\text{BASE,ACTIVE}}_{g,\tau,t}$ is the base power for an active machine, and $\mathrm{E}^{\text{CORE,ACTIVE}}_{g,\tau,t}$ is the incremental energy needed to run a fully loaded core in this power-state. Here E^{MULT} accounts for the typically two- to three-fold increase in energy needed to run power supply units, uninterruptible power supplies, network switches and storage, and most importantly, cooling equipment.

We stipulate the hardware costs for active cores (again omitting the similar expression for idle hardware) as follows:

$$H^{\text{SOLD}} = \sum_{g \in G} \sum_{f \in M_g} \sum_{t \in M_g} \left(\mathrm{H}^{\text{BASE}}_{g,\tau,g} + \mathrm{H}^{\text{TRANSITION}}_{g,f,t}\right) M^{\text{SOLD}}_{g,f,t}$$

where $\mathrm{H}^{\text{BASE}}_{g,\tau,g}$ is the pro-rated cost for each core, and includes not only the amortized server cost, but also supporting equipment, buildings and personnel; and $\mathrm{H}^{\text{TRANSITION}}_{g,f,t}$ accounts for the cost associated within an increased failure rate upon a state transition due to e.g. spinning up/down hard drives. We expect each of these numbers to be easily obtainable through a principled evaluation of existing business practices and capital investments.

Episodic formulations have a common problem in that they may not bear large transition costs when they create a temporary loss, despite a long-term gain. Consequently, we also find it useful to include a predictor on the sell side that tracks the allocation over previous periods (similarly to the buyer demand prediction) and tweaks the optimizers' view of the power state prior to the new period to better match the predicted demand than the actual power state as selected in the previous period. Specifically, we optimistically discount the transition costs for the present time period if the long-term demand level is expected to rise significantly.

A system administrator might, in addition, wish to specify additional restrictions on the allocation to ensure implementability. Considerable flexibility is possible; some examples include: min/max cores/machines for a given application, min/max energy used in a given machine group or for a given application, and max cores in a given machine group that can be allocated to a given application if a certain number are already allocated to specific alternative application (anti-colocation).

6.3 Experimental Results

We have evaluated our market-based system in a set of simulation experiments to show both computational tractability and to show effective allocative behavior over a wider range of environments. Each experiment has been performed on a 3.2GHz dual-processor dual-core workstation with 8GB of memory and CPLEX 11.1. Each data point is the mean of 10 randomly generated time-dependent demand traces.

Our realistic but synthetic traces are the sum of two sinusoid curves (e.g. 1-day period with 9000 peak transactions/minute plus 1-week period with 36000 peak transactions/minute) and a noise term drawn from a Gaussian with a standard deviation equal to 25% of the signal. These match well with real customer traces, where request density is time-dependent and oscillates over both days and weeks.[6] Each transaction is assumed to use 300 MCycles, which is representative of the processing needed to e.g. produce a custom report. Lastly, each allocation period is 10 minutes, which is fast enough to react to dynamic changes in the load but without thrashing.

Because no allocator in the literature has comparable capabilities, we adopt as a benchmark a sophisticated *greedy allocator*, which operates as follows:

1. Put all the machines in their highest efficiency state.
2. Determine the target supply for each application by calculating what is required to produce its ideal response time at its 95^{th} percentile of demand.

[6] Unlike in actual captured data, robustness to load-structure can be tested.

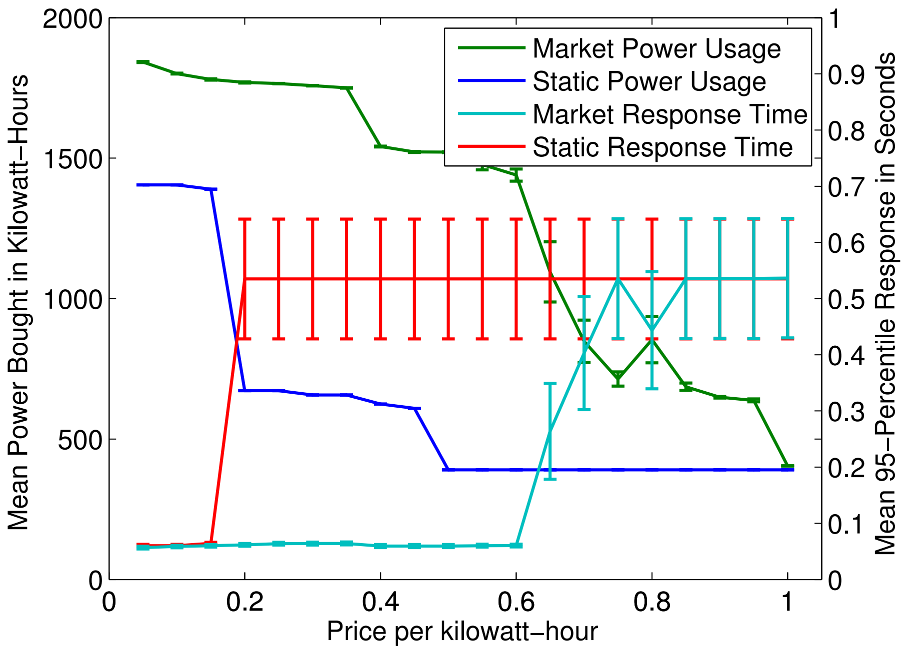


Figure 6.6: Energy used and response time as a function of the price of energy under market and heuristic algorithms.

3. Allocate cores (from feasible machine groups) to the applications, weighted by the marginal value of supply to each application. If an application's supply of high efficiency cores is exhausted, then instead bump one of the machines supporting it into a higher power state. Stop when either all the targets have been met or all the cores/states have been allocated.

4. Consider each application in turn and trim the allocation until the expected value at the 95^{th} percentile of demand is greater than or equal to the expected cost.

5. Place remaining machines in their lowest power state.

For exposition purposes we consider a simple scenario with two applications (i.e. two customers) and three machine groups (each capable of supporting the first, second and both applications respectively), for a simulated week of time-varying demand. Figure 6.6 shows the effect of varying the price of energy under both the market and

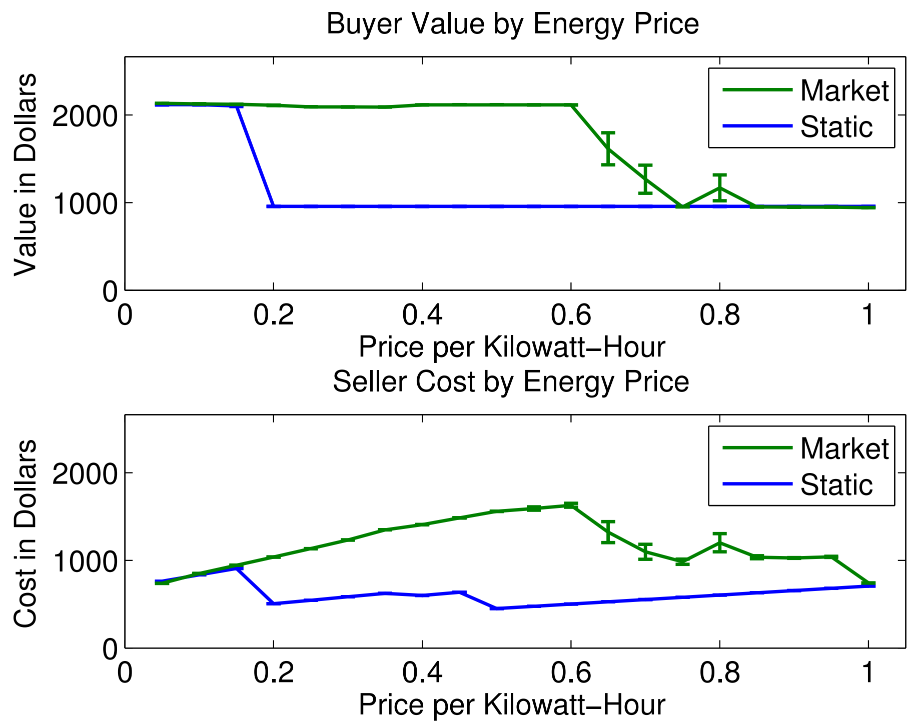


Figure 6.7: Buyer value and seller cost as a function of the price of energy under market and heuristic algorithms.

the static allocation algorithm. We can see that, as expected, under both algorithms the energy used falls and consequently the mean response time rises as the price of energy is increased. However, bidding proxies in the market find it profitable to purchase enough energy to maintain a near-optimal response-time until the price finally reaches such a point that such high energy usage can no longer be sustained, and more energy-frugal allocations are chosen. In Figure 6.7, we see the impact of the choice of these allocations on buyer (i.e. customer) and seller value, as judged by SLAs and revenue net of cost respectively. The greedy allocation is cheaper to provide because of the static power levels, but also results in significantly lower buyer value over a wide range of prices. The maximum revenue net cost improvement is 137% higher in the market model, though margins become slim when energy is expensive.

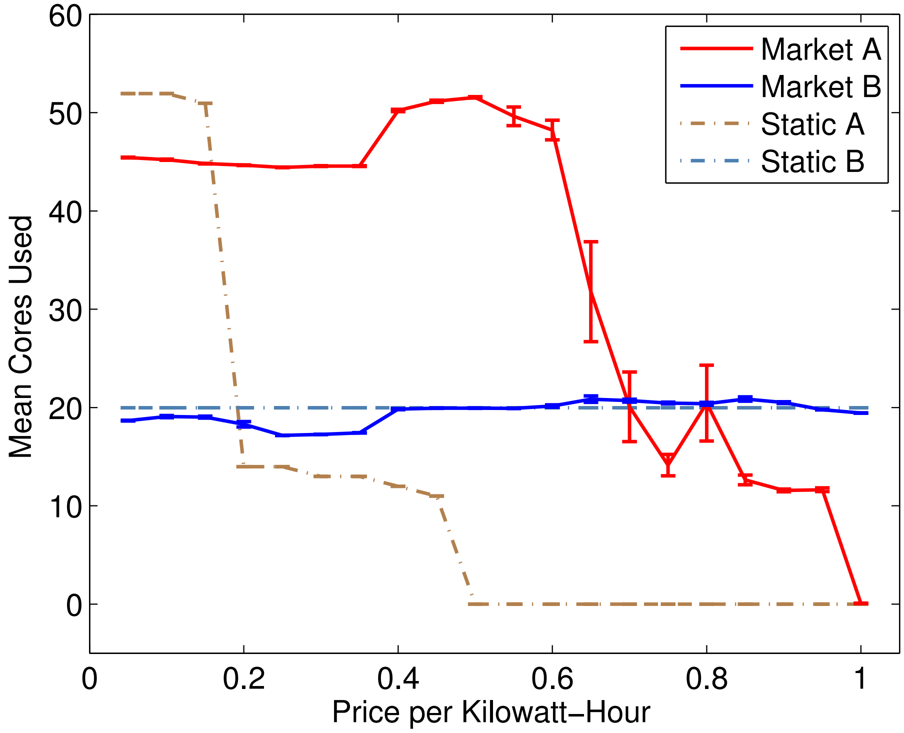


Figure 6.8: Allocation grouped by application as a function of the price of energy under market and heuristic algorithms.

It is also important to consider distributive effects to customers in the data-center setting. In this scenario, the 'A' application induces a larger load then 'B', but with a smaller marginal value for cycles. Consequently, as energy prices rise, the static allocator quickly devotes the limited resources that can be consigned to the 'B' allocation, thereby starving the 'A' application, as seen in Figure 6.8. The market allocation maintains the allocation for the 'B' application, but also recognizes that some resources can profitably be given to 'A'. This is made possible by switching machines between their most efficient modes to conserve energy, and their high-power modes to track spikes in demand. Figure 6.9 shows that in this setting the static allocator has placed all of the machines in the high efficiency 'Low Power' mode, whereas the market has made use of both modes. When the price for power is low,

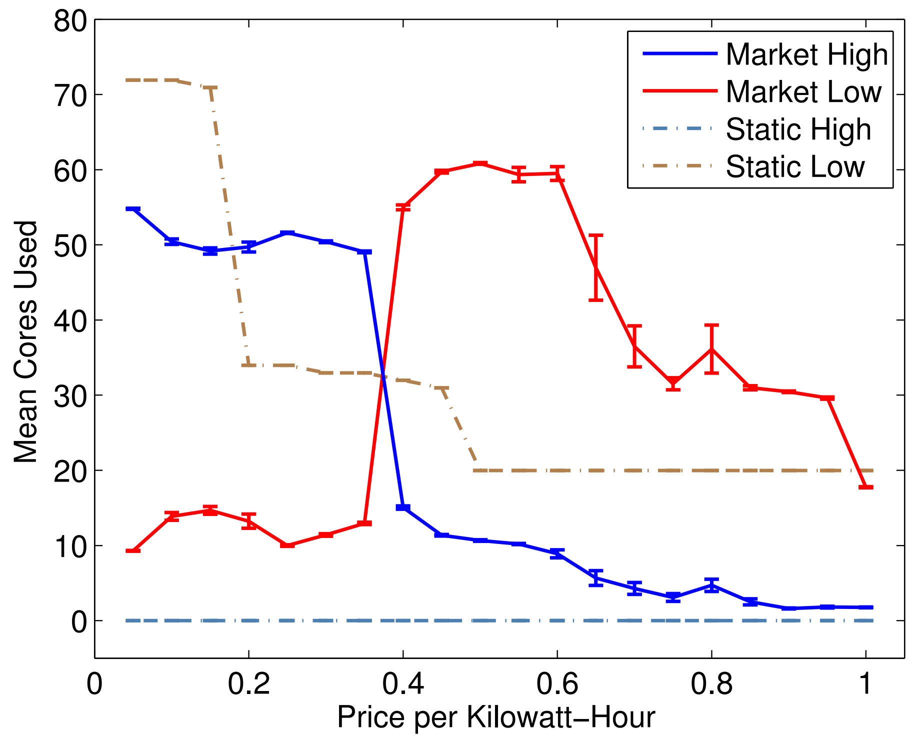


Figure 6.9: Allocation grouped by power mode as a function of the price of energy under market and heuristic algorithms.

the most efficient allocation is to maintain quite a few machines in the high power state. However, as the price crosses 40 cents a kWh, there is a phase change and it becomes much more efficient to run mostly in the low-power mode. Beyond about 60 cents per kWh, it becomes impossible to afford the energy needed to maintain a supply sufficient to keep a low response time, and the optimal allocation shrinks.

6.3.1 Scalability and Complexity

To evaluate the scalability of our MIP formulation we evaluated ten instances of a scenario with 200 quad-core machines in each of five groups, for a total of 1000 machines. We configured ten applications, each with a demand for some 2700 transactions a second, to draw upon these resources with each group supporting three of

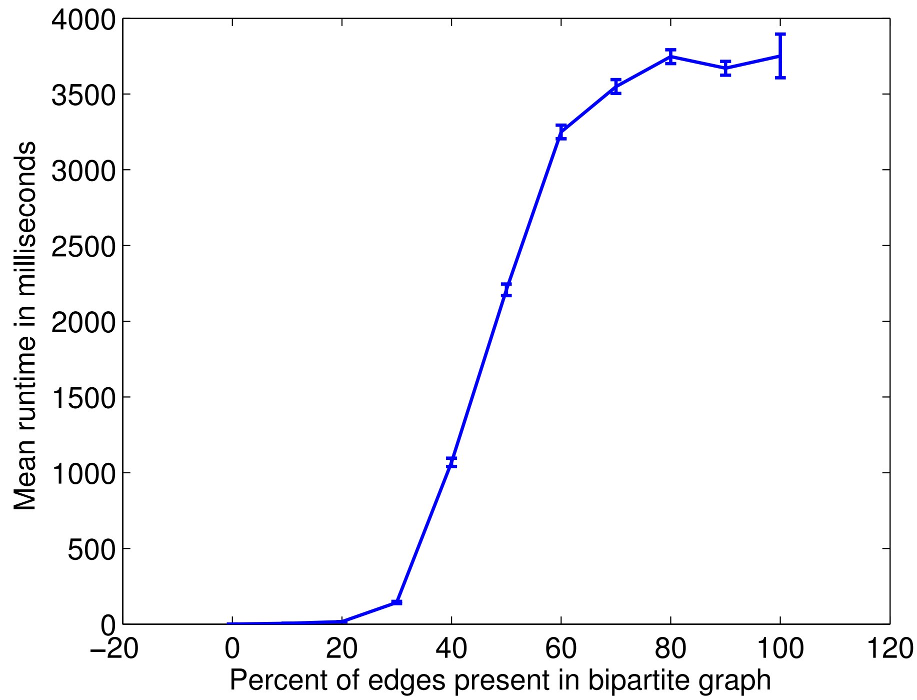


Figure 6.10: Mean runtime by the complexity of the bipartite graph between applications and machine groups.

the applications in a ring topology. We restricted the optimizer to no more then ten minutes of computation per instance, taking advantage of the anytime nature of modern MIP solvers. Including the four instances thus capped, the average solve time was 5.16 minutes, well within the time of a single period. Further, the approximation resulted in only a 0.08% revenue loss when compared to the optimal solution, which would have taken an additional 29 minutes on average for these difficult cases. Thus a single machine is capable of optimizing the usage on 1000 others, giving us a very acceptable 0.1% overhead factor. For a data center with many more machines, one could then decompose them into multiple machine pools, each of size around 1000.

We have also investigated the effect on run-time of the structure of the bipartite graph that defines which application can be supplied by each machine group. For this,

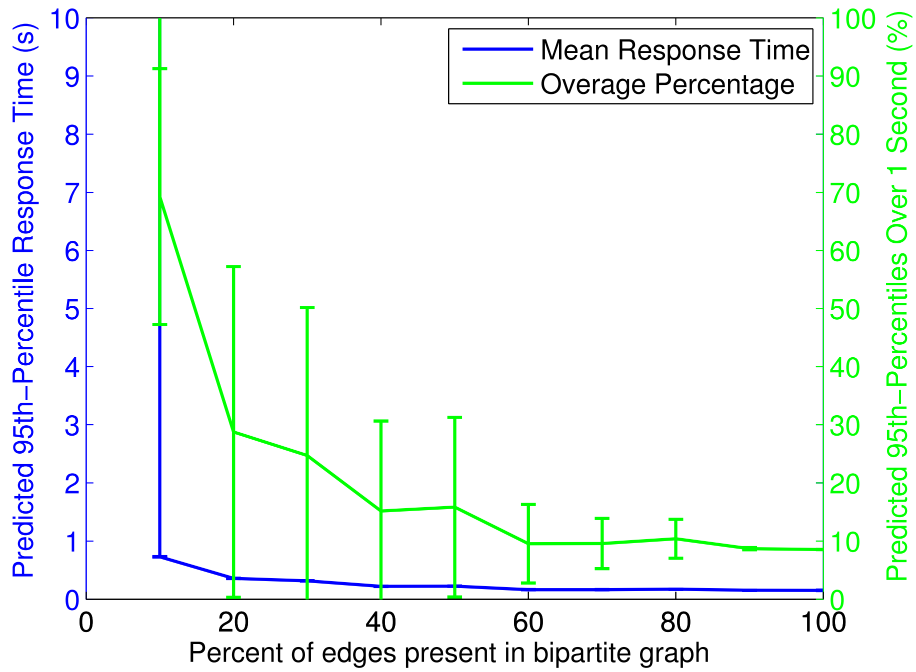


Figure 6.11: 95$^{\text{th}}$ percentile of response time by the complexity of the bipartite graph between applications and machine groups.

we use a scenario with five applications and five machine groups, where supply is set so as to be just sufficient to meet demand. The complexity of the winner-determination problem rises as a sigmoid as we vary the number of edges in the bipartite graph, as shown in Figure 6.10. A graph with 30% of the edges (already highly connected for current data centers) takes only 3.8% of the time needed to clear the market with a complete graph. With 50% connectivity the computation time has risen to 58.8%, and by 60% connectivity the timing has already risen to 86.6%. Further, the increasing complexity is matched by a corresponding decrease in application response time, as shown if Figure 6.11. With 60% of the edges, we are only 8% above the response time of the complete graph.

6.4 Related Work

Chase et al. [2001] present a compelling market-based system for data center resource allocation, and are able to experimentally demonstrate significant energy savings over static allocations. However, their greedy clearing mechanism imposes restrictions on the form of utility that can be modeled, SLAs are not directly represented, and demand/utility computations occur with respect to the mean, not to distributional information. Their model does not handle the heterogeneity of data-center machines or modern power-throttling architectures (instead, simply turning machines on and off), and their allocation is at the level of servers and not cores. The non-linear costing model that we use is related to the one provided by Chen, Das, Qin, Sivasubramaniam, Wang, and Gautam [2005]. But rather than identifying total-value maximizing allocations with respect to SLAs, they treat SLAs as constraints and attempt to find the cheapest allocation subject to meeting implied quality constraints.

Recent work on resource allocation in data centers has focused on *Utility Computing*, which seeks to provide access to the data center in a way analogous to that of a public utility (e.g., gas, water, power) [Byde, 2006; Low and Byde, 2006]. In general, Utility Computing views computational resources as more fungible than in the present work, where we assume that only particular machines are suitably configured for, and capable of running, certain applications. Rather than argue for a more radical shift in how computation is bought, sold, and deployed, in this work we propose a more gradual evolutionary step by creating a market that handles this heterogeneity in present data centers and which encompasses and generalizes the present contract format (SLAs). There is an extensive literature on using market-based methods in related contexts, including *Computational Grids* and in *High Performance Computing*. Yeo and Buyya [2006] and Broberg, Venugopal, and Buyya [2007] are good surveys of this work, which can be informative for the data center domain as well.

6.5 Summary

We have established that suitably designed combinatorial markets can find practical application to power-managed resource allocation in corporate data centers. Further, it is possible to inject revenue-based utility functions directly into the present data center business/allocation model without the large changes associated with Utility Computing, a requirement for rapid industry adoption. Such markets facilitate agent information isolation, quantifying the trade-offs of multi-objective optimization, and facilitate the use of combinatorial optimization in a scalable way, provided carefully-designed models are used.

An experiment is a question which science poses to Nature, and a measurement is the recording of Nature's answer.

– Max Planck
Scientific Autobiography (1949) page 110

7 Conclusion

7.1 Summary

In this thesis we confronted two key problems in the construction of mechanisms for extremely complex settings such as combinatorial auctions:

First, we addressed the issue of how to design around the strong Myerson--Satterthwaite impossibility theorem [Myerson and Satterthwaite, 1983], which states that we cannot simultaneously achieve efficiency, budget balance, individual rationality and incentive-compatibility, even in Bayes-Nash equilibrium and when agents have quasi-linear utilities. We argue that the appropriate response is to relax the incentive compatibility requirement (and thus efficiency as well). Following this agenda, we identified a series of approximate incentive-compatibility measures with which to find mechanisms constructively, or with which to evaluate existing mechanisms tractably.

Second, we addressed the difficulty of agent participation in complex markets and in complex domains. To that end, we designed, implemented, and tested an *iterative* exchange for combinatorial settings. The iteration permits agents to reveal information incrementally and to keep large amounts of their value profile uncertain. We identified a bidding language with straightforward semantics, and carefully constructed easy-to-interpret *linear* prices that help agents interpret how to behave in a complicated economy. Finally we showed how one can apply these concepts to develop a sophisticated market solution to the difficult resource allocation problem of assigning hardware and power in a corporate data center.

7.2 Key Contributions

We highlight the following main contributions of this work:

Defining Approximate Incentive Compatibility: We derived several useful distribution-based criteria for approximate incentive compatibility. The criteria measure the gain that can be achieved under an optimal deviation from truthful reporting under various information assumptions. The conditions are full measures of gain from the BNE, when the distribution is drawn from the equilibrium. But the measures are also well defined for a distribution based on truthful reports of other agents, so we can leverage this simplification in order to avoid having to calculate a full BNE. We also show that the criteria can be used constructively to create optimal mechanisms for simple problems.

Quantiles of *Ex Post* Gain: Of these criteria, the quantile *ex post* stands out as highly informative, while being easy to compute. The 100$^{\text{th}}$ percentile corresponds to the traditional measure of *regret*. But as an absolute worst case analysis, *regret* tells you little about how expectation-maximizing agents will behave. By looking at smaller quantiles, we get far superior information about the likely equilibrium outcome.

KL-Divergence on Payoff Distributions: We offered a computationally simpler way to predict equilibrium behavior: the KL-Divergence between the distribution of payoffs in a given mechanism versus that in a strategyproof "reference" mechanism. We showed that this has a strong positive correlation with the degree to which agents deviate from the truth in a combinatorial exchange setting, along with a strong negative correlation with the efficiency of the consequent outcome. We also offered a theorem that relates this divergence to the *ex ante* expected case criterion that we had developed earlier.

The *TBBL* Language: We presented the *TBBL* bidding language, which permits the concise representation of highly complex valuation functions in a straightforward manner. The ability to have the same item appear in multiple parts of the tree coupled with the semantics of the connective operators is very intuitive; in contrast to the situation with existing logical languages, here one need not examine a distant part of the tree to quantify the value of a given sub-tree. Additionally, the language is capable of directly specifying bounds as is required for the iterative combinatorial exchange design, instead of a fixed value function.

Linear Approximate Competitive Equilibrium Pricing:

In a combinatorial exchange environment, full non-anonymous non-linear prices are needed to establish competitive equilibrium. However, such prices are very unwieldy for agent reasoning. Instead, in the construction of the ICE mechanism, we use linear, anonymous price vectors to drive elicitation. We use a combination of heuristics, lexicographic refinement, and constraint generation to establish a set of unique prices that approximate as closely as possible full competitive equilibrium prices, as well as the payments that will be charged when the exchange closes.

Activity Rules for a Combinatorial Exchange: We propose two new activity rules. The first, MRPAR, is a modified form of the revealed preference rule proposed by Ausubel et al. [2006]. It asks agents to make clear which trade they most prefer at given prices, but degrades gracefully when the linear prices we use are non-discriminating. In this case, we fall back on our second rule, DIAR, which is defined so as to guarantee the ability to drive progress. We prove that together these activity rules will cause the exchange to terminate at the efficient trade with respect to the reported valuations. We also show empirically that agents are not forced to narrow their bounds excessively in order to find this efficient trade.

Trading Performance for Power: Finally, we design a market for allocating resources and energy in a data center environment. We show that by providing tools that can quantify the monetary value for performance, and the true amount (and thus cost) of energy use, we can enable a rational balancing of the competing objectives of high performance and low power consumption.

7.3 Future Work

There are a number of directions for future work that flow out of this thesis. One striking result is that the quantiles of the *ex post* distribution of unilateral incentives to deviate are highly informative about the incentive structure of mechanism equilibrium behavior. This immediately suggests that new metrics that exploit the quantiles of such distributions may be useful. But the agenda is perhaps broader than that. Real agents want a balance between obtaining good behavior in the typical case, and some robustness to bad cases. Quantiles are a very nice way to achieve this: particularly if one has the ability to tailor the shape of the overall quantile curve, within the bounds of feasibility.

Game theory has generally focused on expected-case best responders. However, in the 1970s a few papers considered the use of quantiles in the context of game theory. De Vries [1974] examines strategy choice based on quantile information in two-person matrix games, and a restricted n-player formulation is given by Walsh and Kelleher [1970]. There are known techniques for stochastic optimization using quantiles [Kibzun and Kan, 1996] and for using them in decision theory [Rostek, 2010]. However, there has not been an attempt to apply these approaches to mechanism design. There is potential for some exciting new mechanism design theory that incorporates quantile optimization as a first-class construct. In particular, one might be interested in either or both of two approaches: a) using quantiles as a new solution concept, based on this earlier game theory work, and/or b) finding mechanisms that maximize quantile objectives in closed form. There is also the related goal of finding closed-form mechanisms that are optimal for our simpler KL-Divergence criteria.

Another question that arises out of our work on approximate incentive compatibility is how to deal with the multiple-dimension aspect of the payoff (or gain) distributions. In the work presented here we do a projection down to a single-dimensional setting, and our proof that the KL-Divergence bounds the *ex ante* unilateral gain from deviation is in terms of a single agent. But we know there to be a strong coupling in these distributions—it is precisely this coupling which leads to the exponential space requirements of the LP formulations introduced in Chapter 3. The coupling flows primarily through the budget and core constraints. It would be interesting to see if careful consideration of the true underlying multi-dimension distributions can lead to either improved approximate incentive criteria, or better designs directly.

Another avenue of future work will be to expand the domains for which the types of analysis offered here are performed. It will be interesting to examine settings where the VCG mechanism can still provide the strategyproof benchmark, such as sponsored search with constraints that mandate “simple” payment rules. And, perhaps more challengingly, it would likewise be useful to apply the techniques to domains where the VCG mechanism cannot provide a strategyproof benchmark, for example redistribution mechanisms [Cavallo, 2006].

There are many intriguing opportunities for future work arising out of the ICE mechanism. 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, 1992b] and its generalizations [Babaioff and Walsh, 2005; Chu and Shen, 2008] in domains with restricted expressiveness. We can also consider ICE as a combinatorial *auction* rather than as an exchange, meaning a direct appeal to VCG payments would provide incentive compatibility (except in the presence of core constraints). The other two major directions for future work are to: (a) modify the design to allow bidders to refine the *TBBL* tree *structure* each round, not just their valuation bounds; and (b) extend ICE to work in a dynamic environment with a changing bidder population (e.g. 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, and Zinkevich, 2006; Bredin, Parkes, and 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, there is enormous potential for further work that leverages mechanism design to allocate computation resources. Some topics that warrant further efforts include: richer Service Level Agreement models, hierarchical allocation schemes, distributed implementation of winner allocation schemes, modeling of multiple time scales simultaneously, attempting to combine flexibility of spot markets with the stability of scheduled futures markets, and perhaps most importantly, the ability to balance even more asset types, over and above power and CPU, including bandwidth, memory, and disk access.

7.4 Final Thoughts

Combinatorial markets offer the possibility of significant social benefit, through increases in efficiency. Efficiency gains are possible because the markets offer bidding languages that are expressive enough to capture both participants' non-linear value functions and complex domain specific combinatorial constraints. However, to realize these benefits a design must cope with several difficult problems:

The bidding language needs to be not only expressive but also concise to prevent a requirement for exponential bid size. Moreover, conciseness is not enough on its own since agents may still have too high a cognitive burden if they must determine their entire (exponentially sized) valuation function in order to participate. Iterative mechanisms can help mitigate this problem, by permitting agents to reveal partial information through rounds and by providing price feedback to guide agent reasoning. With a bidding procedure in hand, the center still needs to be able to tractably determine both the winning trade, and the appropriate prices – both of which can be NP-hard problems. Time also plays an important role in design: is the mechanism for a single-shot allocation of goods, or will these goods be reallocated continuously? Are the goods being allocated for an indefinite period of time (spot) or for particular blocks of time (scheduled)?

With the requirement to solve these difficult design problems, there is a tension between general and fully expressive mechanisms (such as ICE), and those which take advantage of domain-specific features to customize the design (as in the last chapter). While the former has wide applicability and re-usability, there is much to gain by specialization.

The efficiency gains that are the draw of combinatorial mechanisms cannot be realized if participants do not provide truthful information to the mechanism. This has justified the central importance of incentive compatible design in the literature, but the construction of such strategyproof mechanisms is generally not possible when we move from auction to exchange settings (or in concert with other desirable features such as core pricing). This motivates the design of the maximally incentive compatible payment rules that were discussed in Chapters 3 and 4. Such rules stand to not only increase efficiency, but just as importantly, to reduce the wasteful agent cognition that arises from the gaming opportunities permitted by simpler approaches.

This thesis has offered both theoretical analysis and concrete proposals for a set of solutions to the myriad practical concerns that arise when implementing these complex mechanism designs; and then applied these lessons to one particular application: power-aware computational resource allocation. There are many other potential applications including bandwidth allocation, airspace contention, advertising sales, and energy markets. While much remains to be learned, the increased efficiency and decreased participatory cost offered by these new mechanisms makes their theoretical and practical development highly worthwhile.

Bibliography

Atila Abdulkadiroglu, Parag A. Pathak, and Alvin E. Roth. The New York City high school match. *American Economic Review, Papers and Proceedings*, 95:364–367, 2005.

Aaron Archer, Christos Papadimitriou, Kunal Talwar, and Éva Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. *Internet Mathematics*, 1(2):129–150, 2004.

Lawrence Ausubel, Peter Cramton, and Paul Milgrom. The Clock-Proxy auction: A practical combinatorial auction design. In Cramton et al. [2006], chapter 5.

Lawrence M. Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1:1–42, 2002.

Lawrence M. Ausubel and Paul Milgrom. The lovely but lonely Vickrey auction. In Cramton et al. [2006], chapter 1.

Moshe Babaioff and William E. Walsh. Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation. *Decision Support Systems*, 39: 123–149, 2005.

Michael Ball, George Donohue, and Karla Hoffman. Auctions for the safe, efficient, and equitable allocation of airspace system resources. In Cramton et al. [2006], chapter 20.

Michael O. Ball, Lawrence M. Ausubel, Frank Berardino, Peter Cramton, George Donohue, Mark Hansen, and Karla Hoffman. Market-based alternatives for managing congestion at New York's LaGuardia airport. In *Proceedings of AirNeth Annual Conference*, 2007.

Salvador Barbera and Matthew O. Jackson. Strategy-proof exchange. *Econometrica: Journal of the Econometric Society*, 63(1):51–87, 1995.

Curt Bererton, Geoff Gordon, and Sebastian Thrun. Auction mechanism design for multi-robot coordination. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS-03)*, 2003.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

Sushil Bikhchandani and John W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74:385–413, 1997.

Sushil Bikhchandani and Joseph M. Ostroy. The package assignment model. *Journal of Economic Theory*, 107(2):377–406, 2002.

Avrim Blum, Tuomas Sandholm, and Martin Zinkevich. Online algorithms for market clearing. *Journal of the ACM*, 53:845–879, 2006.

Craig Boutilier. Solving concisely expressed combinatorial auction problems. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-02)*, pages 359–366, 2002.

Craig Boutilier and Holger Hoos. Bidding languages for combinatorial auctions. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 1121–1217, 2001.

Jonathan Bredin, David C. Parkes, and Quang Duong. Chain: A dynamic double auction framework for matching patient agents. *Journal of Artificial Intelligence Research*, 30:133–179, 2007.

Richard P. Brent. *Algorithms for Minimization Without Derivatives*. Courier Dover Publications, Mineola, NY, 2002. Reprint of Prentice-Hall edition of 1973.

James Broberg, Srikumar Venugopal, and Rajkumar Buyya. Market-oriented grids and utility computing: The state-of-the-art and future directions. *Journal of Grid Computing*, 2007.

Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal income. Technical report, Harvard University, 2009.

Andrew Byde. A comparison between mechanisms for sequential compute resource auctions. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-06)*, pages 1199–1201, 2006.

Andrew Byde, Mathias Salle, and Claudio Bartolini. Market-based resource allocation for utility data centers. Technical Report HPL-2003-188, HP, September 2003.

Ruggiero Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-06)*, 2006.

Jeffrey S. Chase, Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat, and Ronald P. Doyle. Managing energy and server resources in hosting centers. In *Proceedings of the 18th ACM Symposium on Operating Systems and Principles (SOSP-03)*, pages 103–116, 2001.

Yiyu Chen, Amitayu Das, Wubi Qin, Anand Sivasubramaniam, Qian Wang, and Natarajan Gautam. Managing server energy and operational costs in hosting centers. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGmetrics-05)*, pages 303–314, 2005.

Leon Y. Chu and Z. Max Shen. Truthful double auction mechanisms. *Operations Research*, 56(1):102–120, 2008.

Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer, 2001.

Olivier Compte and Philippe Jehiel. Auctions and information acquisition: Sealed-bid or dynamic formats? *Rand Journal of Economics*, 38(2):355–372, 2007.

Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions. In *Proceedings of the 3rd ACM Conference on Electronic Commerce (EC-01)*, pages 256–259, 2001.

Vincent Conitzer. *Computational Aspects of Preference Aggregation*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 2006.

Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 103–110, 2002.

Vincent Conitzer and Tuomas Sandholm. Applications of automated mechanism design. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI-03) Workshop on Bayesian Modeling Applications*, 2003a.

Vincent Conitzer and Tuomas Sandholm. Automated mechanism design: Complexity results stemming from the single-agent setting. In *Proceedings of the 4th ACM Conference on Electronic Commerce (EC-03)*, pages 17–24, 2003b.

Vincent Conitzer and Tuomas Sandholm. Computational criticisms of the revelation principle. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, pages 262–263, 2004a.

Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, page 141, 2004b.

Vincent Conitzer and Tuomas Sandholm. Failures of the VCG mechanism in combinatorial auctions and exchanges. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-06)*, page 528, 2006.

Vincent Conitzer and Tuomas Sandholm. Incremental mechanism design. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1251–1256, 2007.

Florin Constantin and David C. Parkes. On revenue-optimal dynamic auctions for bidders with interdependent values. In *Proceedings of the 9th Workshop on Agent Mediated Electronic Commerce (AMEC-IX)*, Honolulu, HI, 2007.

Peter Cramton. Electricity market design: The good, the bad, and the ugly. In *Proceedings of the Hawaii International Conference on System Sciences*, 2003.

Peter Cramton. Simultaneous ascending auctions. In Cramton et al. [2006], chapter 3.

Peter Cramton, Evan Kwerel, and John Williams. Efficient relocation of spectrum incumbents. *Journal of Law and Economics*, 41:647–675, 1998.

Peter Cramton, Yoav Shoham, and Richard Steinberg, editors. *Combinatorial auctions*. MIT Press, January 2006.

Bernard Dacorogna. *Direct Methods in the Calculus of Variations*. Springer Verlag, New York, 2008.

Robert W. Day and Peter Cramton. The quadratic core-selecting payment rule for combinatorial auctions. Working paper, University of Maryland, 2008.

Robert W. Day and Paul Milgrom. Core-selecting package auctions. *International Journal of Game Theory*, 36(3):393–407, 2008.

Robert W. Day and S. Raghavan. Fair payments for efficient allocations in public sector combinatorial auctions. *Management Science*, 53:1389–1406, 2007.

H. de Vries. Quantile criteria for the selection of strategies in game theory. *International Journal of Game Theory*, 3(2):105–114, 1974.

Sven de Vries and Rakesh V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.

Sven de Vries, James Schummer, and Rakesh V. Vohra. On ascending Vickrey auctions for heterogeneous objects. *Journal of Economic Theory*, 132:95–118, 2007.

Luc Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986.

M. Bernardine Dias, Robert M. Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94:1257–1270, 2006.

Melissa Dunford, Karla Hoffman, Dinesh Menon, Rudy Sultana, and Thomas Wilson. Testing linear pricing algorithms for use in ascending combinatorial auctions. Technical report, SEOR, George Mason University, 2003. Submitted to INFORMS Journal of Computing.

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.

Aytek Erdil and Paul Klemperer. A new payment rule for core-selecting package auctions. *Journal of the European Economic Association*, 8(2–3):537–547, April/May 2010.

David S. Evans. The online advertising industry: Economics, evolution, and privacy. *Journal of Economic Perspectives*, 23(3):37–60, 2009.

Donald F. Ferguson, Christos Nikolaou, Jekka Sairamesh, and Yemini Yemini. Economic models for allocating resources in computer systems. In Scott Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*, pages 156–183. World Scientific Press, 1996.

Yun Fu, Jeffrey Chase, Brent Chun, Stephen Schwab, and Amin Vahdat. SHARP: An architecture for secure resource peering. In *Proceedings of the 19th ACM Symposium on Operating Systems and Principles (SOSP-03)*, pages 133–148, 2003.

Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

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

Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica: Journal of the Econometric Society*, 41:587–602, 1973.

Alison L. Gibbs and Francis E. Su. On choosing and bounding probability metrics. *International Statistical Review/Revue Internationale de Statistique*, pages 419–435, 2002.

Donald B. Gillies. *Some Theorems on n-Person Games*. PhD thesis, Department of Mathematics, Princeton University, 1953.

Jerry R. Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica: Journal of the Econometric Society*, 45:427–438, 1977.

Theodore Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, 41:617–631, 1973.

Mingyu Guo and Vincent Conitzer. Optimal-in-expectation redistribution mechanisms. *Artificial Intelligence*, 174(5–6):363–381, April 2009a.

Mingyu Guo and Vincent Conitzer. Worst-case optimal redistribution of VCG payments in multi-unit auctions. *Games and Economic Behavior*, 67(1):69–98, 2009b.

Mingyu Guo and Vincent Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI-10)*, 2010. to appear.

Bengt Holmström. Groves' scheme on restricted domains. *Econometrica: Journal of the Econometric Society*, 47:1137–1144, 1979.

Benoît Hudson and Tuomas Sandholm. Effectiveness of query types and policies for preference elicitation in combinatorial auctions. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-04)*, pages 386–393, 2004.

Bernard J. Jansen and Tracy Mullen. Sponsored search: An overview of the concept, history, and technology. *International Journal of Electronic Business*, 6(2):114–131, 2008.

Jürgen Jost and Xianqing Li-Jost. *Calculus of variations*. Cambridge University Press, Cambridge, UK, 1998.

Alexander S. Kelso and Vincent P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society*, 50:1483–1504, 1982.

Jeffrey O. Kephart and Rajarshi Das. Achieving self-management via utility functions. *IEEE Internet Computing*, pages 40–48, 2007.

Andreĭ I. Kibzun and Yuri S. Kan. *Stochastic Programming Problems with Probability and Quantile Functions*. Wiley, 1996.

Anshul Kothari, David C. Parkes, and Subhash Suri. Approximately-strategyproof and tractable multi-unit auctions. *Decision Support Systems*, 39:105–121, 2005.

Vijay Krishna. *Auction Theory*. Academic Press, 2002.

Anthony M Kwasnica, John O Ledyard, David Porter, and Christine DeMartini. A new and improved design for multi-object iterative auctions. *Management Science*, 51:419–434, 2005.

Evan Kwerel and John Williams. A proposal for a rapid transition to market allocation of spectrum. Technical report, FCC Office of Plans and Policy, November 2002.

Sébastien Lahaie. Stability and incentive compatibility in a kernel-based combinatorial auction. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI-10)*, 2010. to appear.

Sébastien Lahaie and David C. Parkes. Applying learning algorithms to preference elicitation. In *Proc. 5th ACM Conf. on Electronic Commerce (EC '04)*, pages 180–188. ACM Press, May 2004. Sébastien's thesis is also relevant here.

Sébastien Lahaie, Florin Constantin, and David C. Parkes. More on the power of demand queries in combinatorial auctions: Learning atomic languages and handling incentives. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005.

Sébastien Lahaie, David M. Pennock, Amin Saberi, and Rakesh V. Vohra. Sponsored search auctions. In Cramton et al. [2006].

Kevin Leyton-Brown and Yoav Shoham. A test-suite for combinatorial auctions. In Cramton et al. [2006], chapter 18.

Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, 2000.

Anton Likhodedov and Tuomas Sandholm. Mechanism for optimally trading off revenue and efficiency in multi-unit auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, pages 268–269, 2004a.

Anton Likhodedov and Tuomas Sandholm. Methods for boosting revenue in combinatorial auctions. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04)*, pages 232–237, 2004b.

Anton Likhodedov and Tuomas Sandholm. Approximating revenue-maximizing combinatorial auctions. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, page 267, 2005.

Colin Low and Andrew Byde. Market-based approaches to utility computing. Technical Report HPL-2006-23, HP, Februrary 2006.

Michael Maschler, Bezalel Peleg, and Lloyd S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research*, 4:303–338, 1979.

R. Preston McAfee. A dominant strategy double auction. *Journal of Economic Theory*, 56(2):434–450, 1992a.

R. Preston McAfee. A dominant strategy double auction. *Journal of Economic Theory*, 56:434–450, 1992b.

Paul Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004.

Paul Milgrom. Package auctions and package exchanges (2004 Fisher-Schultz lecture). *Econometrica: Journal of the Econometric Society*, 75:935–966, 2007.

Paul Milgrom and Robert W. Day. Core-selecting auctions. *International Journal of Game Theory*, 36:393–407, 2008.

Debasis Mishra and David C. Parkes. Ascending price Vickrey auctions for general valuations. *Journal of Economic Theory*, 132:335–366, 2007.

S. Muthukrishnan. Ad exchanges: Research issues. *Intenet and Network Economics*, pages 1–12, 2009.

Roger B. Myerson. Incentive compatibility and the bargaining problem. *Econometrica: Journal of the Econometric Society*, 47:61–73, 1979.

Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6: 58–73, 1981.

Roger B. Myerson and Mark A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265–281, 1983.

John Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.

Roger B. Nelsen. *An Introduction to Copulas*. Springer, New York, 1999.

George Nemhauser and Laurence Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.

Noam Nisan. Bidding languages for combinatorial auctions. In Cramton et al. [2006], chapter 9.

Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, pages 242–252, 2000.

Richard P. O'Neill, Paul M. Sotkiewicz, Benjamin F. Hobbs, Michael H. Rothkopf, and William R. Stewart, Jr. Efficient market-clearing prices in markets with non-convexities. *European Journal of Operations Research*, 164:269–285, 2005.

Abraham Othman and Tuomas Sandholm. Better with byzantine: Manipulation-optimal mechanisms. In Marios Mavronicolas and Vicky Papadopoulou, editors, *Algorithmic Game Theory*, volume 5814 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 2009.

David C. Parkes. Optimal auction design for agents with hard valuation problems. In *Proceedings of the IJCAI-99 Workshop on Agent Mediated Electronic Commerce*, pages 206–219, Stockholm, Sweden, July 1999.

David C. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, May 2001.

David C. Parkes. Auction design with costly preference elicitation. *Annals of Mathematics and Artificial Intelligence*, 44:269–302, 2005. Special Issue on the Foundations of Electronic Commerce.

David C. Parkes. Online mechanisms. In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*, chapter 16. Cambridge University Press, 2007.

David C. Parkes. When analysis fails: Heuristic mechanism design via self-correcting procedures. In *Proceedings of the 35th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-09)*, 2009.

David C Parkes and Lyle H Ungar. Iterative combinatorial auctions: Theory and practice. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 74–81, 2000a.

David C Parkes and Lyle H Ungar. Preventing strategic manipulation in iterative auctions: Proxy agents and price-adjustment. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 82–89, 2000b.

David C. Parkes and Lyle H. Ungar. An ascending-price generalized Vickrey auction. In *Proceedings of the Stanford Institute for Theoretical Economics Workshop on The Economics of the Internet*, Stanford, CA, 2002.

David C. Parkes, Jayant R Kalagnanam, and Marta Eso. Achieving budget-balance with Vickrey-based payment schemes in exchanges. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 1161–1168, 2001a.

David C. Parkes, Jayant R. Kalagnanam, and Marta Eso. Achieving budget-balance with Vickrey-based payment schemes in combinatorial exchanges. Technical report, IBM Research Report RC 22218, 2001b. Updated, March 2002.

Parag A. Pathak and Tayfun Sönmez. Comparing mechanisms by their vulnerability to manipulation. working paper, January 2009.

Chris Preist, Andrew Byde, Claudio Bartolini, and Giacomo Piccinelli. Towards agent-based service composition through negotiation in multiple auctions. Techincal Report HPL-2001-71R1, HP, May 2002.

Steve J. Rassenti, Vernon L. Smith, and R. L. Bulfin. A combinatorial mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982.

Daniel M. Reeves and Michael P. Wellman. Computing best-response strategies in infinite games of incomplete information. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 470–478, 2004.

Ori Regev and Noam Nisan. The POPCORN market: Online markets for computational resources. *Decision Support Systems*, 28(1-2):177–189, 2000.

Marzena Rostek. Quantile maximization in decision theory. *Review of Economic Studies*, 77(1):339–371, 2010.

Michael H. Rothkopf. Thirteen reasons why the vickrey-clarke-groves process is not practical. *Operations Research*, 55(2):191–197, 2007.

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

Kemal Saatcioglu, Jan Stallaert, and Andrew B. Whinston. Design of a financial portal. *Communications of the ACM*, 44:33–38, 2001.

Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93)*, page 256, 1993.

Tuomas Sandholm. Issues in computational vickrey auctions. *International Journal of Electronic Commerce*, 4(3):129, 2000.

Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002a.

Tuomas Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18(4):656–676, 2002b.

Tuomas Sandholm. Optimal winner determination algorithms. In Cramton et al. [2006], chapter 14.

Tuomas Sandholm. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3): 45, 2007.

Tuomas Sandholm and Craig Boutilier. Preference elicitation in combinatorial auctions. In Cramton et al. [2006], chapter 10.

D Schmeidler. The nucleolus of a characteristic function game. *Society for Industrial and Applied Mathematics Journal of Applied Mathematics*, 17:1163–1170, 1969.

James Schummer. Almost dominant strategy implementation. Technical report, MEDS Department, Kellogg Graduate School of Management, 2001.

Lloyd S. Shapley and Martin Shubik. The assignment game I: The core. *International Journal of Game Theory*, 1:111–130, 1972.

Trey Smith, Tuomas Sandholm, and Reid Simmons. Constructing and clearing combinatorial exchanges using preference elicitation. In *18th National Conference on Artificial Intelligence (AAAI-02) Workshop on Preferences in AI and CP: Symbolic Approaches*, 2002.

György Steinbrecher and William T. Shaw. Quantile mechanics. *European Journal of Applied Mathematics*, 19(2):87–112, 2008.

Malgorzata Steinder, Ian Whalley, James E. Hanson, and Jeffrey O. Kephart. Coordinated management of power usage and runtime performance. In *Proceedings of the 9th International Symposium on Integrated network Management (NOMS-08)*, pages 387–394, 2008.

Ivan E. Sutherland. A futures market in computer time. *Communications of the ACM*, 11(6):449–451, 1968.

DOE. *Carbon Dioxide Emissions from the Generation of Electric Power in the United States*. US DOE and EPA, Jul 2000.

EIA. *Emissions of Greenhouse Gases Report: Carbon Diaoxide Emissions*. US EIA, Dec 2008.

EPA. *Report to Congress on Server and Data Center Energy Efficiency*. US EPA, Aug 2007.

William Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

Cédric Villani. *Optimal Transport Old and New*. Springer, Berlin, 2009. Grundlehren der mathematischen Wissenschaften, 338.

Yevgeniy Vorobeychik and Michael P. Wellman. Stochastic search methods for Nash equilibrium approximation in simulation-based games. In Burg Berger and Nishiyama, editors, *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-08)*, Estoril, Portugal, May 2008.

Yevgeniy Vorobeychik, Christopher Kiekintveld, and Michael P. Wellman. Empirical mechanism design: Methods, with application to a supply chain scenario. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC-06)*, pages 306–315, 2006.

Yevgeniy Vorobeychik, Daniel M. Reeves, and Michael P. Wellman. Constrained automated mechanism design for infinite games of incomplete information. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI-07)*, 2007.

Thomas W. M. Vossen and Michael O. Ball. Slot trading opportunities in collaborative ground delay programs. *Transportation Science*, 40:15–28, 2006.

Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and W. Scott Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2):103–117, 1992.

John E. Walsh and Grace J. Kelleher. Generally applicable n-person percentile game theory for case of independently chosen strategies. Technical Report 92, Southern Methodist University Department of Statistics, Dallas Texas, December 1970.

Steven R. Williams. A characterization of efficient, bayesian incentive compatabile mechanism. *Economic Theory*, 14:155–180, 1999.

Peter R. Wurman and Michael P. Wellman. Equilibrium prices in bundle auctions. In *Proceedings of the Workshop on Artificial Intelligence for Electronic Commerce (AAAI-99)*, pages 56–61, 1999.

Peter R. Wurman and Michael P. Wellman. A*k*BA: A progressive, anonymous-price combinatorial auction. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, pages 21–29, 2000.

Chee S. Yeo and Rajkumar Buyya. A taxonomy of market-based resource management systems for utility-driven cluster computing. *Software, Practice & Experience*, 36(13):1381–1419, 2006.

Makoto Yokoo. Pseudonymous bidding in combinatorial auctions. In Cramton et al. [2006], chapter 7.

Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. Effect of false-name bids in combinatorial auctions: New fraud in internet auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.

Makoto Yokoo, Vincent Conitzer, Tuomas Sandholm, Naoki Ohta, and Atsushi Iwasaki. Coalitional games in open anonymous environments. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, page 509, 2005.

Gilad Zlotkin and Jeffrey S. Rosenschein. Mechanism design for automated negotiation, and its application to task oriented domains. *Artificial Intelligence*, 86(2): 195–244, 1996.