Chapter 1

AUCTIONS, BIDDING AND EXCHANGE DESIGN

Jayant Kalagnanam
IBM Research Division
T.J.Watson Research Center,
P.O.Box 218,
Yorktown Heights, NY 1058
jayant@us.ibm.com

David C. Parkes
Division of Engineering and Applied Sciences,
Harvard University,
Cambridge MA 02138
parkes@eecs.harvard.edu

Abstract *The different auction types are outlined using a classification framework along six dimensions. The economic properties that are desired in the design of auction mechanisms and the complexities that arise in their implementation are discussed. Some of the most interesting designs from the literature are analyzed in detail to establish known results and to identify the emerging research directions.*

1. Introduction

Auctions have found widespread use in the last few years as a technique for supporting and automating negotiations on the Internet. For example, eBay now serves as a new selling channel for individuals, and small and big enterprises. Another use for auctions is for industrial procurement. In both these settings traditional auction mechanisms such as the English, Dutch, First (or Second) price Sealed-Bid auctions are now commonplace. These auctions types are useful for settings where there is a single unit of an item being bought/sold. However, since procurement problems are business-to-business they tend to be more complex and have led to the development and application of advanced auction types that allow for negotiations over multiple units of multiple items, and the configuration of the attributes of items. At the heart of auctions is the problem of decentralized resource allocation.

A general setting for decentralized allocation is one with multiple agents with utility functions for various resources. The allocation problem for the decision maker, or *intermediary*, is to allocate these resources in an optimal way. A key difference from the classical optimization perspective is that the utility function of the agents is private information, and not explicitly known to the decision maker. In addition, standard methods in decentralized optimization fail because of the self-interest of participants. Therefore the design of decentralized allocation mechanisms must provide incentives for agents to reveal their true preferences in order to solve for the optimal allocation with respect to the true utility functions. Thus, the behavioral aspects of agents must be explicitly considered in the design. It is common in the economic mechanism design literature to assume rational, game-theoretic, agents. Another common assumption is that agents behave as myopic price-takers, that are rational in the current round of negotiation but not necessarily with respect to the final outcomes at the end of the negotiation.

In settings where the allocation problem itself is hard even if the decision maker knows the "true" utility function of each agent, the issues of incentive compatibility makes the design of an appropriate auction mechanism even more challenging.

The focus of this chapter is to provide an overview of the different auction mechanisms commonly encountered both in practice and in the literature. We will initially provide a framework for classifying auction mechanisms into different types. We will borrow a systems perspective (from the literature) to elucidate this framework.

1.1 A framework for auctions

We develop a framework for classifying auctions based on the requirements that need to be considered to set up an auction. We have identified these core components below:

Resources The first step is to identify the set of resources over which the negotiation is to be conducted. The resource could be a single item or multiple items, with a single or multiple units of each item. An additional consideration common in real settings is the type of the item, i.e. is this a standard commodity or multiattribute commodity. In the case of multiattribute items, the agents might need to specify the non-price attributes and some utility/scoring function to tradeoff across these attributes.

Market Structure An auction provides a mechanism for negotiation between buyers and sellers. In *forward auctions* a single seller is selling resources to multiple buyers. Alternately, in *reverse auctions*, a single buyer is sourcing resources from multiple suppliers, as is common in procurement. Auctions with multiple buyers and sellers are called *double auctions* or *exchanges*, and these are commonly used for trading securities and financial instruments and increasingly within the supply chain.

Preference Structure The preference structure of agents in an auction is important and impacts some of the other factors. The preferences define an agent's utility for different outcomes. For example, when negotiating over multiple units agents might indicate a decreasing marginal utility for additional units. An agent's preference structure is important when negotiation over attributes for an item, for designing scoring rules used to signal information.

Bid Structure The structure of the bids allowed within the auction defines the flexibility with which agents can express their resource requirements. For a simple single unit, single item commodity, the bids required are simple statements of willingness to pay/accept. However, for a multi-unit identical items setting bids need to specify price and quantity. Already this introduces the possibility for allowing volume discounts, where a bid defines the price as a function of the quantity. With multiple items, bids may specify all-or-nothing bids with a price on a basket of items. In addition, agents might wish to provide several alternative bids but restrict the choice of bids.

Matching Supply to Demand A key aspect of auctions is matching supply to demand, also referred to as market clearing, or *winner determination*. The main choice here is whether to use *single-sourcing*, in which pairs of buyers and sellers are matched, or *multi-sourcing*, in which multiple suppliers can be matched with a single buyer, or vice-versa. The form of matching influences the complexity of winner determination, and problems range the entire spectrum from simple sorting problems to NP-hard optimization problems.

Information Feedback Another important aspect of an auction is whether the protocol is a *direct* mechanism or an *indirect* mechanism. In a direct mechanism, such as the first price sealed bid auction, agents submit bids without receiving feedback, such as price signals, from the auction. In an indirect mechanism, such as an ascending-price auction, agents can adjust bids in response to information feedback from the auction. Feedback about the state of the auction is usually characterized by a *price signal* and a *provisional allocation*, and provides sufficient information about the bids of other agents to enable an agent to refine its bids. In complex settings, such as multi-item auctions with bundled bids, a direct mechanism can require an exponential number of bids to specify an agent's preference structure. In comparison, indirect mechanisms allow incremental revelation of preference information, on a "as required basis". The focus in the design of indirect mechanisms is to identify how much preference information is sufficient to achieve desired economic properties and how to implement informationally-efficient mechanisms. A related strand of research is to provide compact bidding languages for direct mechanisms.

Each of the six dimensions that we have identified provide a vector of choices that are available to set up the auction. Putting all of these together generates a matrix of auction types. The choices made for each of these dimensions will have a major impact on the complexity of the analysis required to characterize the market structure that emerges, on the complexity on agents and the intermediary to implement the mechanism, and ultimately on our ability to design mechanisms that satisfy desirable economic and computational properties.

1.2 Outline

In this chapter we first introduce the economic literature on *mechanism design*, and identify the economic properties that are desirable in the design of auction mechanisms. Then, in Section 3, we introduce the associated computational complexities that arise in the implementation of optimal mechanisms, and discuss tradeoffs that must often be made between optimality and computational tractability. We will not attempt to describe the sum total of all known mechanisms that are available in the literature. Instead, in Section 4 we pick a few mechanisms that are interesting both from a practical point of view and also because they illustrate some of the emerging research directions. Finally, Section 5 provides an overview of experimental approaches to analyzing economic behavior, and suggests an interesting direction in *automated mechanism design* for electronic markets.

2. Economic Considerations

The basic economic methodology used in the design of electronic intermediaries first models the preferences, behavior, and information available to agents, and then designs a mechanism in which agent strategies result in outcomes with desirable properties. We consider two approaches to modeling agent behavior:

game-theoretic/mechanism design The first model of agent behavior is *game-theoretic* and relates to *mechanism design theory*. In this model the equilibrium state is defined by the condition that agents play a best-response strategy to each other and cannot benefit from a unilateral deviation to an alternative strategy.

price-taking/competitive equilibrium The second model of agent behavior is *price-taking*, or *myopic best-response*, and relates to *competitive equilibrium theory*. In this model the equilibrium state is defined by the condition that an agent plays a best-response to the *current* price and allocation in the market, without modeling either the strategies of other agents or the effect of its own actions on the future state of the market.

Mechanism design theory and game-theoretic modeling is most relevant when one or both of the following conditions hold: (a) the equilibrium solution concept makes weak game-theoretic assumptions about agent behavior, such as when a mechanism can be designed with a *dominant strategy* equilibrium, in which agents have a single strategy that is always optimal *whatever* the strategies and preferences of other agents; or (b) there are a small number of agents and it is reasonable to expect agents to be rational and well-informed about the likely preferences of other agents. Competitive equilibrium theory and price-taking modeling is most relevant in large systems in which the effect of an agent's own strategy on the state of a market is small, or when there is considerable uncertainty about agent preferences and behaviors and no useful mechanism with a dominant strategy equilibrium.

2.1 Preliminaries

Our presentation is limited to the *private value* model, in which the value to an agent for an outcome is only a function of its own private information. This is quite reasonable in the procurement of goods for direct consumption, unless there are significant opportunities for resale or unless there is significant uncertainty about the quality of goods. *Correlated* and *common* value models may be more appropriate in these settings, and the prescriptions for mechanism design can change [PMM87].

Consider $\mathcal{I} = (1, \ldots, N)$ agents, a discrete outcome space $\mathcal{K}$, and payments $p = (p_1, \ldots, p_N) \in \mathbb{R}^N$, where p_i is the payment from agent i to the mechanism. The private information associated with agent i, which defines its value for different outcomes, is denoted with *type*, $\theta_i \in \Theta_i$. Given type θ_i, then agent i has value $v_i(k, \theta_i) \in \mathbb{R}$ for outcome $k \in \mathcal{K}$. It is useful to use $\theta = (\theta_1, \ldots, \theta_N)$ to denote a type vector, and $\Theta = \Theta_1 \times \ldots \times \Theta_N$ for the joint type space. In simple cases in which an agent's valuation function can be represented by a single number, for example in a single-item allocation problem, it is convenient to write $v_i = \theta_i$.

We assume risk neutral agents, with *quasilinear utility functions*, $u_i(k, p, \theta_i) = v_i(k, \theta_i) - p$. This is a common assumption across the auction and mechanism design literature. Although an agent knows its own type, there is incomplete information about the types of other agents. Let $f_i(\theta_i) \in [0, 1]$ denote the probability density function over the type, θ_i, of agent i, and let $F_i(\theta_i) \in [0, 1]$ denote the corresponding cumulative distribution function. We assume that the types of the agents are independent random variables, and that there is *common knowledge* of these distributions, such that agent i knows $f_j(\cdot)$ for every other agent $j \neq i$, agent j knows that agent i knows, etc. We assume that the mechanism designer has the same information as the agents.

2.2 Mechanism Design

The mechanism design approach to solving distributed allocation problems with self-interested agents formulates the design problem as an *optimization* problem. Mechanism design addresses the problem of implementing solutions to distributed problems despite the fact that agents have private information about the quality of different solutions and that agents are self-interested and happy to misreport their private information if that can improve the solution in their favor. A mechanism takes information from agents and makes a decision about the outcome and payments that are implemented. It is useful to imagine the role of a mechanism designer as that of a game designer, able to determine the rules of the game but not the strategies that agents will follow.

A mechanism defines a set of feasible *strategies*, which restrict the kinds of messages that agents can send to the mechanism, and makes a commitment to use a particular *allocation rule* and a particular *payment rule* to select an outcome and determine agent payments, as a function of their strategies.[1] Game theoretic methods are used to analyze the properties of a mechanism, under the assumption that agents are rational and will follow expected-utility maximizing strategies in equilibrium.

Perhaps the most successful application of mechanism design has been to the theory of auctions. In recent years auction theory has been applied to the design of a number of real-world markets [Mil02]. There are two natural design goals in the application of mechanism design to auctions and markets. One goal is *allocative efficiency*, in which the mechanism implements a solution that maximizes the total payoff across all agents. This is the *efficient* mechanism design problem. Another goal is *payoff maximization*, in which the mechanism implements a solution that maximizes the payoff to a particu-

[1] A mechanism must be able to make a commitment to use these rules. Without this commitment ability the equilibrium of a mechanism can quickly unravel. For example, if an auctioneer in a second-price auction cannot commit to selling the item at the second-price than the auction looks more like a first-price auction [PMM87].

lar agent. This is the *optimal* mechanism design problem. One can imagine many other variations, including settings in which the goal is to maximize the total payoff across a subset of agents, or settings in which the fairness of an allocation matters.

In particular settings, such as when there is an efficient after-market, then the optimal mechanism is also an efficient mechanism [AC98], but in general there exists a conflict between efficiency and optimality [Mye81]. Competition across marketplaces can also promote goals of efficiency, with the efficient markets that maximize the total payoff surviving in the long-run [HRN02]. Payoff maximization for a single participant is most appropriate in a setting in which there is asymmetric market power, such as in the automobile industry when market power within the supply chain is held by the big manufacturers [Che93, BW01].

The efficient mechanism design problem has proved more tractable than the optimal mechanism design problem, with optimal payoff-maximizing mechanisms known only in quite restrictive special cases.

2.2.1 Direct Revelation Mechanisms. The space of possible mechanisms is huge, allowing for example for multiple rounds of interaction between agents and the mechanism, and for arbitrarily complex allocation and payment rules. Given this, the problem of determining the *best* mechanism from the space of all possible mechanisms can appear impossibly difficult. The *revelation principle* [Gib73, GJJ77, Mye81] allows an important simplification. The revelation principle states that it is sufficient to restrict attention to *incentive compatible direct-revelation* mechanisms. In a direct-revelation mechanism (DRM) each agent is simultaneously asked to report its type. In an incentive-compatible (IC) mechanism each agent finds it in their own best interest to report its type truthfully. The mechanism design problem reduces to defining functions that map types to outcomes, subject to constraints that ensure that the mechanism is incentive-compatible. To understand the revelation principle, consider taking a complex mechanism, $\mathcal{M}$, and constructing a DRM, $\mathcal{M}'$, by taking reported types and simulating the equilibrium of mechanism $\mathcal{M}$. If a particular strategy, $s^*(\theta)$, is in equilibrium in $\mathcal{M}$, given types θ, then truthful reporting of types is in equilibrium in $\mathcal{M}'$ because this induces strategies $s^*(\theta)$ in the simulated mechanism.

Care should be taken in interpreting the revelation principle. First, the revelation principle does not imply that “incentive-compatibility comes for free”. In fact, a central theme of mechanism design is that there is a cost to the elicitation of private information. The mechanism design literature is peppered with impossibility results that characterize sets of desiderata that are impossible to achieve simultaneously because it is necessary to incent agents to participate in a mechanism [Jac00]. Rather, the revelation principle states that *if* a particular set of properties can be implemented in the equilibrium of some mechanism, then the properties can also be implemented in an incentive-compatible mechanism. Second, the revelation principle ignores computation and communication complexity, and should not be taken as a statement that “only direct revelation mechanisms matter in practical mechanism design”. In many cases indirect mechanisms are preferable for reasons unmodeled in classic mechanism design theory, for example because they decentralize computation to participants, and can economize on preference elicitation while achieving more transparency than direct mechanisms. We return to this topic in Section 3.

The beauty of the revelation principle is that it allows theoretical impossibility and possibility results to be established in the space of direct mechanisms, and carried over to apply to *all* mechanisms. For example, an indirect mechanism can be constructed with a particular set of properties only if a direct mechanism can also be constructed with the same set of properties.

2.2.2 Efficient Mechanism Design. In efficient mechanism design, the goal is to implement the choice, $k^* \in \mathcal{K}$, that maximizes that total value across all agents given agent types, $\theta \in \Theta$. By the revelation principle we can focus on incentive-compatible DRMs. Each agent is asked to report its type, $\hat{\theta}$, possibly untruthfully, and the mechanism chooses the outcome and the payments. The mechanism defines an allocation rule, $g : \Theta \rightarrow \mathcal{K}$, and a payment rule, $p : \Theta \rightarrow \mathbb{R}^N$. Given reported types, $\hat{\theta}$, then choice $g(\hat{\theta})$ is implemented and agent i makes payment $p_i(\hat{\theta})$.[2]

The goal of efficiency, combined with incentive-compatibility, pins down the allocation rule:

$$g_{\text{eff}}(\theta) = \arg\max_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} v_i(k, \theta_i) \tag{EFF}$$

for all $\theta \in \Theta$. The remaining mechanism design problem is to choose a payment rule that satisfies IC, along with any additional desiderata. Popular additional criteria include:

(IR) individual-rationality. An agent’s expected payoff is greater than its payoff from non-participation.

(BB) budget-balance. Either *strong*, such that the total payments made by agents equal zero, or *weak*, such that the total payments made by agents are non-negative.

(revenue) maximize the total expected payments by agents.

[2] Later, in discussion of optimal mechanism design, we will fall back on the more general framework of randomized allocation rules and expected payments. For now we choose to stick with deterministic allocations and payments to keep the notation as simple as possible.

Given payment rule, $p(\cdot)$, and allocation rule, $g(\cdot)$, let $m_i(p, \hat{\theta}_i)$, $V_i(g, \hat{\theta}_i \mid \theta_i)$, and $U_i(g, p, \hat{\theta}_i \mid \theta_i)$ denote (respectively) the expected payment, expected valuation, and expected payoff to agent i when reporting type, $\hat{\theta}_i$, assuming the other agents are truthful. It is convenient to leave the dependence of $m_i(\cdot)$ on $g(\cdot)$ and the dependence of $V_i(\cdot)$ on $p(\cdot)$ implicit.

$$m_i(p, \hat{\theta}_i) = E_{\theta_{-i}}[p_i(\hat{\theta}_i, \theta_{-i})] \qquad \text{(}\textit{interim}\text{ payment)}$$

$$V_i(g, \hat{\theta}_i \mid \theta_i) = E_{\theta_{-i}}[v_i(g(\hat{\theta}_i, \theta_{-i}), \theta_i)] \qquad \text{(}\textit{interim}\text{ valuation)}$$

$$U_i(g, p, \hat{\theta}_i \mid \theta_i) = V_i(g, \hat{\theta}_i \mid \theta_i) - m_i(p, \hat{\theta}_i) \qquad \text{(}\textit{interim}\text{ payoff)}$$

Notation $\theta_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_N)$ denotes the type vector without agent i. The expectation is taken with respect to the joint distribution over agent types, θ_{-i}, implied by marginal probability distribution functions, $f_i(\cdot)$. Assuming IC, then $m_i(p, \theta_i)$, $V_i(g, \theta_i \mid \theta_i)$ and $U_i(g, p, \theta_i \mid \theta_i)$ are the expected payment, valuation, and payoff to agent i in equilibrium. These are also referred to as the *interim* payments, valuations, and payoffs, because they are computed once an agent knows its own type but before it knows the types of the other agents. It is often convenient to suppress the dependence on the specific mechanism rules (g, p) and write $m_i(\theta_i)$, $V_i(\theta_i)$ and $U_i(\theta_i)$. Finally, let $m_i(p) = E_{\theta_i}[m_i(p, \theta_i)]$ denote the expected *ex ante* payment by agent i, before its own type is known.

The efficient mechanism design problem is formulated as an optimization problem across payment rules that satisfy IC, as well as other constraints such as IR and BB. These constraints define the space of *feasible* payment rules. A *selection criteria*, $y(m_1, \ldots, m_N) \in \mathbb{R}$, defined over expected payments, can be used to choose a particular payment rule from the space of feasible rules. A typical criteria is to maximize the total expected payments, with $y(m_1, \ldots, m_N) = \sum_i m_i$. Formally, the efficient mechanism design problem [EFF] is:

$$\max_{p(\cdot)} \quad y(m_1(p), \ldots, m_N(p)) \qquad \text{[EFF]}$$

$$\text{s.t.} \quad U_i(g_{\text{eff}}, p, \theta_i \mid \theta_i) \geq U_i(g_{\text{eff}}, p, \hat{\theta}_i \mid \theta_i), \quad \forall i, \forall \theta_i \in \Theta_i \qquad \text{(IC)}$$

$$\text{additional constraints} \qquad \text{(IR),(BB),etc.}$$

where $g_{\text{eff}}(\cdot)$ is the efficient allocation rule.

The IC constraints require that when other agents truthfully report their types an agent's best response is to truthfully report its own type, for all possible types. In technical terms, this ensures that truth-revelation is a Bayesian-Nash equilibrium, and we say that the mechanism is *Bayesian-Nash* incentive-compatible. In a *Bayesian-Nash* equilibrium every agent is plays a strategy that is an expected utility maximizing response to its beliefs over the *distribution* over the strategies of other agents. An agent need not play a best-response to the *actual* strategy of another agent, given its actual type. This equilibrium is strengthened in a dominant strategy equilibrium, in which truth-revelation is the best-response for an agent whatever the strategies and preferences of other agents. A dominant strategy and IC mechanism is simply called a *strategyproof* mechanism. Formally:

$$v_i(g(\theta_i, \theta_{-i}), \theta_i) - p(\theta_i, \theta_{-i}) \geq v_i(g(\hat{\theta}_i, \theta_{-i}), \theta_i) - p(\hat{\theta}_i, \theta_{-i}), \quad \forall i, \forall \theta_i, \forall \theta_{-i} \tag{SP}$$

Strategyproofness is a useful property because agents can play their equilibrium strategy without game-theoretic modeling or counterspeculation about other agents.

Groves [Gro73] mechanisms completely characterize the class of efficient and strategyproof mechanisms [GJJ77]. The payment rule in a Groves mechanism is defined as:

$$p_{\text{groves},i}(\hat{\theta}) = h_i(\hat{\theta}_{-i}) - \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}))$$

where $h_i(\cdot) : \Theta_{-i} \rightarrow \mathbb{R}$ is an arbitrary function on the reported types of every agent except i, or simply a constant. To understand the strategyproofness of the Groves mechanisms, consider the utility of agent i, $u_i(\hat{\theta}_i)$, from reporting type $\hat{\theta}_i$, given $g_{\text{eff}}(\cdot)$ and $p_{\text{groves}}(\cdot)$, and fix the reported types, θ_{-i}, of the other agents. Then, $u_i(\hat{\theta}_i) = v_i(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_i) - p_{\text{groves},i}(\hat{\theta}_i, \theta_{-i})$, and substituting for $p_{\text{groves}}(\cdot)$, we have $u_i(\hat{\theta}_i) = v_i(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_j) - h_i(\theta_{-i})$. Reporting $\hat{\theta}_i = \theta_i$ maximizes the sum of the first two terms by construction, and the final term is independent of the reported type. This holds for all θ_{-i}, and strategyproofness follows. The Groves payment rule internalizes the externality placed on the other agents in the system by the reported preferences of agent i. This aligns an agent's incentives with the system-wide goal of allocative-efficiency.

The uniqueness of Groves mechanisms provides an additional simplification to the efficient mechanism design problem when *dominant strategy* implementations are required. It is sufficient to consider the family of Groves mechanisms, and look for functions $h_i(\cdot)$ that provide Groves payments that satisfy all of the desired constraints. The Vickrey-Clarke-Groves (VCG) mechanism is an important special case, so named because it reflects the seminal ideas due to Vickrey [Vic61] and Clarke [Cla71]. The VCG mechanism maximizes expected revenue across all strategyproof efficient mechanisms, subject to *ex post* individual-rationality (IR) constraints. *Ex post* IR provides:

$$v_i(g(\theta_i, \theta_{-i}), \theta_i) - p_i(\theta_i, \theta_{-i}) \geq 0, \quad \forall i, \forall \theta_i, \forall \theta_{-i} \tag{\textit{ex post} IR}$$

This is an *ex post* condition, because it requires that the equilibrium payoff to an agent is always non-negative at the outcome of the mechanism, whatever the types of other agents. To keep things simple we assume that an agent has zero payoff for non-participation. The VCG mechanism defines payment:

$$p_{\mathrm{vcg},i}(\hat{\theta}) = \sum_{j \neq i} v_j(g_{\mathrm{eff}}(\hat{\theta}_{-i})) - \sum_{j \neq i} v_j(g_{\mathrm{eff}}(\hat{\theta}))$$

where $g_{\mathrm{eff}}(\hat{\theta}_{-i})$ is the efficient allocation as computed with agent i removed from the system.

It is natural to ask whether greater revenue can be achieved by relaxing strategyproofness to Bayesian-Nash IC. In fact, the VCG mechanism maximizes the expected revenue across *all* efficient and *ex post* IR mechanisms, even allowing for Bayesian-Nash implementation [KP98]. This equivalence result yields a further simplification to the efficient mechanism design problem, beyond that provided by the revelation principle. Whenever the additional constraints (in addition to IR and IC) are *interim* or *ex ante* in nature[3] in an efficient mechanism design problem, then it is sufficient to consider the family of Groves mechanisms in which the arbitrary $h_i(\cdot)$ functions are replaced with constants [Wil99]. Not only is the allocation rule, $g(\cdot)$, pinned down, but so is the functional form of the payment rule, $p(\cdot)$, and the mechanism design problem reduces to optimization over a set of constants.

This analysis of the revenue properties of VCG mechanisms follows from a fundamental *payoff equivalence* result [KP98, Wil99]. The payoff equivalence result states that

$$U_i(\theta_i) = U_i(\underline{\theta}_i) + \int_C \left. \frac{\mathrm{d}V_i(\theta_i)}{\mathrm{d}\theta_i} \right|_{\theta_i = \tau} \mathrm{d}\tau \qquad \text{(equiv)}$$

for all efficient mechanisms, where $\underline{\theta}_i$ is the minimal type of agent i, and C is a smooth curve from $\underline{\theta}_i$ to θ_i within Θ_i. By definition (*interim* valuation), the *interim* valuation, $V_i(\theta_i)$, in an IC mechanism depends only on the allocation rule. Therefore payoff equivalence (equiv) states that the equilibrium payoff from any two IC mechanisms with the same allocation rule, $g(\cdot)$, are equal up to an additive constant, i.e. its payoff at some particular type $\underline{\theta}_i$. A consequence of payoff equivalence is that all IC mechanisms with the same allocation rule are *revenue equivalent* up to an additive constant, which is soon pinned down by additional constraints such as IR.[4]

[3] *Ex ante* and *interim* refer to timing within the mechanism. *Ex ante* constraints are defined in expectation, before agent types are known. *Interim* constraints are defined relative to the type of a particular agent, but in expectation with respect to the types of other agents.

[4] As a special case, we get the celebrated *revenue-equivalence theorem* [Vic61, Mye81], which states that the most popular auction formats, i.e. English, Dutch, first-price sealed-bid and second-price sealed-bid,

Finally, this characterization of the VCG mechanism provides a unified perspective on many areas of mechanism design theory, and provides a simple and direct proof of a number of impossibility results in the literature [KP98]. As an example, we can consider the Myerson-Satterthwaite [MS83] impossibility result, which demonstrates a conflict between efficiency and budget-balance in a simple two-sided market. There is one seller and one buyer, a single item to trade, and agent preferences such that both no-trade and trade can be efficient *ex ante*. There does not exist an efficient, weak budget-balanced, and IR mechanism in this setting and any efficient exchange with voluntary participation must run at a budget deficit. Recalling that the VCG mechanism maximizes expected payments from agents across all efficient and IR mechanisms, there is a simple constructive method to prove this negative result. One simply shows that the VCG mechanism in this setting runs at a deficit.

2.2.3 Optimal Mechanism Design.

In optimal mechanism design the goal is to maximize the expected payoff of *one* particular agent. Recall that the primary goal in efficient mechanism design is to maximize the total payoff across *all* agents. The agent receiving this special consideration in the context of optimal auction design is often the *seller*, although this need not be the role of the agent. We find it convenient to refer to this agent as the seller in our discussion, and indicate this special agent with index 0. In optimal mechanism design the goals of the designer are aligned with the seller, and it is supposed that we have complete information about the seller's type. The mechanism design problem is formulated over the remaining agents, to maximize the expected payoff of the seller subject to IR constraints.

Myerson [Mye81] first introduced the problem of optimal mechanism design, in the context of an auction for a single item with a seller that seeks to maximize her expected revenue. We will provide a general formulation of the optimal mechanism design problem, to parallel the formulation of the efficient mechanism design problem. However, *analytic* solutions to the optimal mechanism design problem are known only for special cases.

In this section we allow randomized allocation and payment rules. The allocation rule, $g : \Theta \rightarrow \Delta(\mathcal{K})$, defines a probability distribution over choices given reported types, and the payment rule, $p : \Theta \rightarrow \mathbb{R}^N$, defines expected payments. The ability to include non-determinism in the allocation rule allows the mechanism to break ties at random, amongst other things. Let $V_0(g, p)$ denote the expected *ex ante* valuation of the seller for the outcome, in equilibrium given the payment and allocation rules and beliefs about agent types.

all yield the same price on average in a single item allocation problem with symmetric agents. This is an immediate consequence because these auctions are all efficient in the simple private values model.

By the revelation principle we can restrict attention to IC DRMs, and immediately express the optimal mechanism design problem [OPT] as

$$\max_{g(\cdot),p(\cdot)} \quad V_0(g,p) + \sum_i m_i(p) \qquad \text{[OPT]}$$

$$\begin{aligned} \text{s.t.} \quad & U_i(g,p,\theta_i \mid \theta_i) \geq U_i(g,p,\hat{\theta}_i \mid \theta_i), \quad \forall i, \forall \theta_i \in \Theta_i && \text{(IC)} \\ & \text{additional constraints} && \text{(IR),(BB),etc.} \end{aligned}$$

where $m_i(p)$ is the expected equilibrium payment made by agent i, $U_i(g,p,\hat{\theta}_i \mid \theta_i)$ is the expected equilibrium payoff to agent i with type θ_i for reporting type $\hat{\theta}_i$. The objective is to maximize the payoff of the seller. In comparison with the efficient mechanism design problem, we have no longer pinned down the allocation rule and the optimization is performed over the entire space of allocation and payment rules.

One approach to compute an optimal mechanism is to decompose the problem into a master problem and a subproblem. The subproblem takes a particular allocation rule, $g'(\cdot)$, and computes the optimal payment rule given $g'(\cdot)$, subject to IC constraints. The masterproblem is then to determine an allocation rule to maximize the value of the subproblem. However, as discussed by Vohra & de Vries in Chapter 4, the set of allocation rules need not be finite or countable, and this is a hard problem without additional structure. Solutions are known for special cases, including a single-item allocation problem [Mye81], and also a simple multiattribute allocation problem [Che93].

As an illustration, we provide an overview of optimal mechanism design for the single-item allocation problem. Let $\pi_{g,i}(\hat{\theta}) \geq 0$ denote the probability that agent i receives the item, given reported types $\hat{\theta}$ and allocation rule $g(\cdot)$. We also write, $v_i(k_i, \theta_i) = \theta_i$, for the choice, k_i, in which agent i receives the item, and 0 otherwise, so that an agent's type corresponds to its value for the item. Let θ_0 denote the seller's value.

Call a mechanism (g,p) *feasible* if IC and *interim* IR hold. The first step in the derivation of the optimal auction reduces IC and *interim* IR to the following conditions on (g,p):

$$Q_i(g,\theta_1) \leq Q_i(g,\theta_2), \quad \forall i \in \mathcal{I}, \forall \theta_1 < \theta_2, \forall \theta_1, \theta_2 \in \Theta_i \qquad (1.1)$$

$$U_i(g,p,\theta_i) = U_i(g,p,\underline{\theta}_i) + \int_{\tau=\underline{\theta}_i}^{\theta_i} Q_i(g,\tau)\mathrm{d}\tau, \quad \forall i \in \mathcal{I}, \forall \theta_i \in \Theta_i \qquad (1.2)$$

$$U_i(g,p,\underline{\theta}_i) \geq 0, \quad \forall i \in \mathcal{I} \qquad (1.3)$$

where $\underline{\theta}_i$ represents the lowest possible value that i might assign to the item, and $Q_i(g,\hat{\theta}_i)$ denotes the conditional probability that i will get the item when reporting type, $\hat{\theta}_i$, given that the other agents are truthful, i.e. $Q_i(g,\hat{\theta}_i) = E_{\theta_{-i}}[\pi_{g,i}(\hat{\theta}_i, \theta_{-i})]$.

The key to this equivalence is to recognize that IC can be expressed as:

$$U_i(g,p,\theta_i \mid \theta_i) \geq U_i(g,p,\hat{\theta}_i \mid \hat{\theta}_i) + (\theta_i - \hat{\theta}_i)Q_i(g,\hat{\theta}_i), \quad \forall \hat{\theta}_i \neq \theta_i \qquad (1.4)$$

in this single-item allocation problem by a simple substitution for $U_i(g,p,\hat{\theta}_i \mid \theta_i)$. Given this, condition (1.1), which states that an agent's probability of getting the item must decrease if it announces a lower type, together with (1.2) implies condition (1.4), and IR follows from (1.2) and (1.3).

Continuing, once the payoff to an agent with type $\underline{\theta}$ is pinned down, then the interim payoff (1.2) of an agent is independent of the payment rule because $Q_i(g,\tau)$ is the conditional probability that agent i receives the item given type τ and allocation rule g. This allows the optimal mechanism design problem to be formulated as an optimization over just the *allocation rule*, with the effect of computing an optimal solution to the payoff-maximizing subproblem for a given allocation rule folded into the masterproblem, and IR constraints allowing the seller's expected payoff to be expressed in terms of the expected payoffs of the other agents. Integration of Q_i between $\underline{\theta}_i$ and θ_i yields a simplified formulation:

$$\max_{g(\cdot)} E_\theta \left[\sum_{i \in \mathcal{I}} (J_i(\theta_i) - \theta_0)\pi_{g,i}(\theta) \right] \qquad \text{[OPT']}$$

$$\text{s.t.} \quad Q_i(g,\theta_1) \leq Q_i(g,\theta_2), \quad \forall i \in \mathcal{I}, \forall \theta_1 < \theta_2, \forall \theta_1, \theta_2 \in \Theta_i \qquad (1.1)$$

where the value, $J_i(\theta_i)$, is the *priority level* of agent i, and computed as:

$$J_i(\theta_i) = \theta_i - \frac{1 - F_i(\theta_i)}{f_i(\theta_i)}$$

Recall that $f_i(\cdot)$ is the probability distribution over the type of agent i, and $F_i(\cdot)$ the cumulative distribution. This priority level, sometimes called the *virtual valuation*, is less than an agent's type by the expectation of the second-order statistic of the distribution over its type. Economically, one can imagine that this represents the "information rent" of an agent, the expected payoff that an agent can extract from the private information that it has about its own type.

The optimal allocation rule, $g_{\text{opt}}(\cdot)$, requires the seller to *keep the item if* $\theta_0 > \max_i J_i(\hat{\theta}_i)$ *and award it to the agent with the highest* $J_i(\hat{\theta}_i)$ *otherwise, breaking ties at random.* It is immediate that this rule maximizes the objective [OPT']. A technical condition, *regularity*, ensures that this allocation rules satisfies (1.1). Regularity requires that the priority, $J_i(\theta_i)$, is a monotone strictly increasing function of θ_i for every agent. Myerson [Mye81] also derives a general solution for the non-regular case. The remaining problem, given g_{opt}, is to solve for the payment rule. The optimal payment rule given a particular allocation rule is computed as:

$$p_i(\theta) = \pi_{g,i}(\theta)\theta_i - \int_{\tau=\underline{\theta}_i}^{\theta_i} \pi_{g,i}(\tau, \theta_{-i})\mathrm{d}\tau \tag{1.5}$$

where $\pi_{g,i}(\theta)$ is the probability that i gets the item given g and types θ. Given allocation rule, g_{opt}, this simplifies to

$$p_i(\theta) = \begin{cases} \inf\{\hat{\theta}_i \mid J_i(\hat{\theta}_i) \geq \theta_0, J_i(\hat{\theta}_i) \geq J_j(\theta_j), \forall j \neq i\} & \text{, if } \pi_{g_{\mathrm{opt}},i}(\theta) = 1 \\ 0 & \text{, otherwise.} \end{cases}$$

where θ_0 is the value of the seller for the item. In words, only the winner makes a payment, and the payment is the smallest amount the agent could have bid and still won the auction. This payment rule makes truth-revelation a Bayesian-Nash equilibrium of the auction.

The optimal auction is a Vickrey auction with a reservation price in the special case that all agents are symmetric and the $J_i(\cdot)$ functions are strictly increasing. The seller places a reservation price, $p_0 = J^{-1}(\theta_0)$, given her value, θ_0, and the item is sold to the highest bidder for the second-highest price whenever the highest bid is greater than the reservation price. The optimal auction in this symmetric special case is strategyproof. The effect of the seller's reservation price is to increase the payment made whenever the seller's price is between the second-highest and highest bid from outside bidders, at the risk of missing a trade when the highest outside bid is lower than the seller's reservation price but higher than the seller's true valuation. Notice that the optimal auction is not *ex post* efficient.

In the general case of asymmetric bidders the optimal auction may not even sell to the agent whose value for the item is the highest. In this asymmetric case the optimal auction is *not* a Vickrey auction with a reservation price. The agent with the highest priority level gets the item, and the effect of adjusting for the prior beliefs $f_i(\cdot)$ about the type of an agent is that the optimal auction discriminates against bidders that *a priori* are expected to have higher types. This can result in an agent with a higher type having a lower priority level than an agent with a lower type. One can imagine that the optimal auction price-discriminates across buyers based on beliefs about their types.

2.3 Competitive Equilibrium

Competitive equilibrium theory is built around a model of agent price-taking behavior. At its heart is nothing more than linear-programming duality theory. One formulates a primal problem to represent an efficient allocation problem, and a dual problem to represent a pricing problem. Competitive equilibrium conditions precisely characterizes complementary-slackness conditions between an allocation and a set of prices, and implies that the allocation is optimal and therefore efficient. Competitive equilibrium conditions are useful because they can be evaluated based on myopic best-response bid information from agents, and without requiring complete information about agent valuations. This is the sense in which prices can decentralize decision-making in resource allocation problems.

The modeling assumption of price-taking behavior states that agents will take prices as given and demand items that maximize payoff given their valuations and the current prices. This is commonly described as *price-taking* or *myopic best-response* behavior. In the language of mechanism design, this can be considered a form of *myopic*, or bounded, incentive-compatibility. In some problems, there are competitive equilibrium (CE) prices that also implement the VCG payments [BO02], and this myopic assumption is no longer required. Indeed, an iterative auction that terminates with CE prices corresponding with VCG payments inherits incentive-compatibility properties from the VCG mechanism. The connection between linear programming, competitive equilibrium, and VCG payments has been used to design ascending-price auctions to implement the outcome of the VCG mechanism in a number of problems [Par01, PU02, BdVSV01, DGS86]. The methodology is discussed in Section 3.5, and a number of examples are provided in Section 4.

To illustrate CE prices we will impose some structure on choice set $\mathcal{K}$. Let $\mathcal{G}$ define a set of items, and $S \subset \mathcal{G}$ a subset, or *bundle*, of items. A choice, $k \in \mathcal{K}$ defines a feasible allocation of bundles to agents. Introduce variables, $x_i(S) \in \{0,1\}$, to indicate that agent i receives bundle S in a particular allocation. In addition, it is convenient to express an agent's preference structure as a valuation function over bundles, $v_i(S)$, such that $v_i(k, \theta_i) = \sum_{S \subseteq \mathcal{G}} v_i(S) x_i(S)$, for $\sum_{S \subseteq \mathcal{G}} x_i(S) \leq 1$.

Given information about agent valuations, $v_i(\cdot)$, the efficient allocation problem can be formulated as an integer program:

$$\begin{aligned} \max_{x_i(S)} \sum_{S \subseteq \mathcal{G}} \sum_{i \in \mathcal{I}} & x_i(S) v_i(S) \\ \text{s.t.} \quad \sum_{S} x_i(S) &\leq 1, \quad \forall i \in \mathcal{I} \\ \sum_{S \ni j} \sum_{i \in \mathcal{I}} x_i(S) &\leq 1, \quad \forall j \in \mathcal{G} \\ x_i(S) &\in \{0,1\} \end{aligned}$$

where $S \ni j$ indicates that bundle S contains item j.

To apply linear-programming duality theory we must relax this IP formulation, and construct an integral LP formulation. Consider [LP$_1$] and its dual program, [DLP$_1$]:

$$\max_{x_i(S)} \sum_S \sum_i x_i(S) v_i(S) \qquad [\mathrm{LP}_1]$$

$$\text{s.t.} \quad \sum_S x_i(S) \leq 1, \qquad \forall i \in \mathcal{I} \qquad (\mathrm{LP}_1\text{-}1)$$

$$\sum_{S \ni j} \sum_i x_i(S) \leq 1, \qquad \forall j \in \mathcal{G} \qquad (\mathrm{LP}_1\text{-}2)$$

$$x_i(S) \geq 0, \qquad \forall i, S$$

$$\min_{\pi_i, p(j)} \sum_i \pi_i + \sum_j p(j) \qquad [\mathrm{DLP}_1]$$

$$\text{s.t.} \quad \pi_i + \sum_{j \in S} p(j) \geq v_i(S), \qquad \forall i \in \mathcal{I}, \forall S \subseteq \mathcal{G} \qquad (\mathrm{DLP}_1\text{-}1)$$

$$\pi_i, p(j) \geq 0, \qquad \forall i, j$$

The dual problem introduces variables $p(j) \geq 0$, for items $j \in \mathcal{G}$, which we can interpret as prices on items. Given prices, $p(j)$, the optimal dual solution sets $\pi_i = \max_S \left\{ v_i(S) - \sum_{j \in S} p(j), 0 \right\}$. This is the maximal payoff to agent i given the prices, and the dual problem computes prices on items to minimize the sum of the payoffs across all agents.

The dual solution computes CE prices when the primal solution is integral. A technical condition, *gross substitutes* [KC82], on agent valuations is sufficient for integrality. Given gross substitutes, then complementary-slackness (CS) conditions on a feasible primal, x, and feasible dual, p, solution define conditions for competitive equilibrium:

$$\pi_i > 0 \Rightarrow \sum_S x_i(S) = 1, \quad \forall i \qquad (1.6)$$

$$p(j) > 0 \Rightarrow \sum_{S \ni j} \sum_i x_i(S) = 1, \quad \forall j \qquad (1.7)$$

$$x_i(S) > 0 \Rightarrow \pi_i + \sum_{j \in S} p(j) = v_i(S), \quad \forall i, \forall S \qquad (1.8)$$

Conditions (1.6) and (1.8) state that the allocation must maximize the payoff for every agent at the prices. Condition (1.7) states that the seller must sell every item with a positive price, and maximize the payoff to the seller at the prices. When these conditions hold then prices are CE and the allocation is efficient. Notice that CE prices provide a *certificate* for efficiency. A seller can announce an efficient allocation and CE prices, and let every agent verify that the allocation maximizes its own payoff at the prices.

Without gross substitutes it is necessary to strengthen the LP formulation to achieve integrality and compute CE prices. One can construct a hierarchy of formulations [BO02], with duals that compute *non-linear* and then non-linear and *non-anonymous* prices. Non-linear prices, $p(S) \geq 0$, on bundles $S \subseteq \mathcal{G}$, allow $p(S) \neq p(S_1)+p(S_2)$ for $S = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$. Non-anonymous prices, $p_i(S) \geq 0$, on bundles S to agent i, allow $p_i(S) \neq p_j(S)$ for $i \neq j$.

In Section 4.1 we use primal-dual theory to derive an ascending-price *combinatorial auction* for an extended formulation, in which the dual problem computes non-linear prices. The combinatorial auction allows agents to express their preferences by submitting bids on bundles of items.

3. Implementation Considerations

In this section, we discuss some of the computational considerations that must be addressed in taking a mathematical specification of a mechanism and building a working system. There can often remain a large gap between the mathematical specification and a reasonable computational implementation. In this section we lay out some of the details that must be considered in closing this gap.

Particular implementation considerations include the choice of a language to represent agent preferences (Section 3.1) and the complexity of the winner-determination problem (Section 3.2), which can also be impacted by side constraints that represent business rules (Section3.4). Sometimes it is necessary to implement an *indirect* variation of the direct revelation mechanism, for example in problems in which the direct mechanism requires an unreasonable amount of preference information from agents (Section 3.5). Sometimes the situation is worse, and there is no reasonable implementation of the optimal mechanism. In such cases, computational considerations must be introduced explicitly during the mechanism design process itself (Section 3.6).

3.1 Bidding Language

The structure of the bidding language in an auction is important because it can restrict the ability of agents to express their preferences. In addition, the expressiveness allowed also has a big impact of the the properties of the auction. This has prompted research that examines bidding languages and their expressiveness and the impact on winner determination [BH01, Bou02, BK02]. In this section we will outline two aspects of bidding languages that are central to auctions: (i) the structure of bids allowed, and (ii) the rules specified by the bid that restrict the choice of bids by the seller.

The structure of bids that are allowed are closely related to the market structure. For example, in markets where multiple units are being bought or sold it becomes necessary to allows bids that express preferences over multiple units. Some common bid structures examined in the literature are:

- divisible bids with price-quantity pairs that specify per-unit prices and allow any amount less than specified quantity can be chosen.
- divisible bids with a price schedule, for example volume discounted bids
- indivisible bids with price-quantity pairs, where the price is for the total amount bid and this is to be treated as an all-or-nothing bid.
- bundled bids with price-quantity pairs, where the bid is indivisible and the price is over the entire basket of different items and is to be treated as an all-or-nothing bid.
- configurable bids for multiattribute items that allow the bidder to specify a bid function sensitive to attribute levels chosen.

With multiple items or multiattribute items the preference structure of agents can be exponentially large. For example, if there are n items and the agent has super-additive preferences then in general the agent could specify 2^n bids. Multiattribute items with n binary attributes leads to similar informational complexity. Therefore an additional consideration is to provide a compact bid representation language that allows agents to implicitly specify their bid structure. Several researchers have proposed mechanisms for specifying bids logically. Boutilier and Hoos [BH01] provide a nice overview of logical bidding languages for combinatorial auctions. These bidding languages have two flavors: (i) logical combinations of goods as formulae ($\mathbb{L}_G$), and (ii) logical combinations of bundles as formulae ($\mathbb{L}_B$).

$\mathbb{L}_G$ [BH01, HB00] languages allow bids that are logical formulae where goods (items) are taken as atomic propositions and combined using logical connectives and a price is attached to the formula expressing the amount that the bidder is willing to pay for satisfaction of this formula. $\mathbb{L}_G$ captures perfect substitutes with disjunctions in a single formula, however imperfect substitutes might require multiple formulae to capture the agent's preferences. $\mathbb{L}_B$ [San00, Nis00] language uses bundles with associated prices as atomic propositions and combines them using logical connectives. Sandholm [San00] suggested $\mathbb{L}_B^{OR}$ that uses disjunctions over atomic bids. Semantically, these languages are interpreted by assigning goods to the component atoms and the price is determined as the sum of the prices of the atomic bids that are satisfied. Nisan [Nis00] considered $\mathbb{L}_B^{XOR}$ allowing exclusive-OR and two-level nesting of such connectives which allows expression of substitutability. Another alternative is to allow dummy goods within atomic bids to make the language more compact. Nisan [NR00] provides a discussion of the relative merits of these languages. More recent work [BH01] introduced $\mathbb{L}_{GB}$ for generalized logical bids that allows a combination of both items and bundles as atomic propositions within a single formula. This inherits the advantages of both approaches and allows concise specification of utility.

Similar issues of concise representation of preferences over multiattribute items/goods is explored in Bichler et al [BKL02]. A bid can specify the values that are allowed for each attribute and an associated markup price over the base levels. In addition, an atomic proposition is associated with each value for each attribute and horn clauses are used to specify configurations that are not allowed or to specify promotions associated with certain feature sets.

3.2 Winner-Determination Complexity

Two primary components of a mechanism are the *allocation rule* and the *payment rule*. As discussed in Section 2.2, two objectives considered in mechanism design are allocative-efficiency and optimality with respect to the payoff to a particular agent. In order to implement an allocation rule for either of these objectives an optimization problem needs to be solved. This optimization problem is typically referred to as the *winner determination* problem. In simple designs (such as the English, Dutch etc) where only a single winner is permitted in the allocation, the optimization problem can be solved in a straightforward fashion. However, in settings where the allocation rule permits multiple winners, the optimization problem that needs to be solved can become quite computationally complex depending on the market and bid structures. In this section we outline the different settings and the associated complexity of the winner determination problem.

3.2.1 Multi-Unit Auctions. Consider an auction for multiple units of the same type of item, and in particular the reverse auction setting where the focus is to minimize the cost subject to bid requirements. We will consider three cases: (i) divisible bids, (ii) indivisible bids with XOR bid structures and (iii) price schedules which can be viewed as a compact representation for generalized XOR indivisible bids.

Suppose that a buyer requests to buy Q identical units of the same item. The bidders respond with a bid B_i, defining price-quantity pair (p_i, q_i). In the simple case where the bids are **divisible**, the optimal allocation can simply be identified by sorting the bids in increasing order of unit price and picking the cheapest bids until the demand for Q is satisfied. In general, the last chosen bid might get a partial allocation. However, the following bid structures lead to winner determination problems that require solving optimization problems:

Indivisible Bids. If the bidders specify all-or-nothing constraints on the bids then the bids are indivisible. Let M_i denote the number of bids from supplier i, and N denote the number of suppliers. The winner determination problem can be formulated as a knapsack problem, introducing $x_{ij} \in \{0,1\}$ to indicate that bid j from bidder i is accepted [BK02].

$$\begin{aligned} \min_{x_{ij}} \sum_{i=1}^{N}\sum_{j=1}^{M_i} p_{ij}x_{ij} \\ \text{s.t.} \quad \sum_{i=1}^{N}\sum_{j=1}^{M_i} q_{ij}x_{ij} \geq Q \\ \sum_{j=1}^{M_i} x_{ij} \leq 1, \quad \forall i \\ x_{ij} \in \{0,1\} \end{aligned}$$

The special case where each bidder has a single bid reduces to a knapsack problem which is NP-hard [MT80]. In order to write this as a knapsack problem use the transformation $y_{ij} = 1 - x_{ij}$ and rewrite the formulation as a maximization problem.

Price Schedules. If the bids incorporate price schedules (such as volume discounts) then the winner determination can be modeled as a generalization of the multiple choice knapsack problem. The key issue is whether the price schedule is nonlinear or piecewise linear. Piecewise linear approximations are commonly used to model nonlinear functions [DK01, SS01b]. Therefore, we will focus on a model with piecewise linear price schedules.

Each supplier responds with a price schedule that consists of a list of M_i price quantity pairs, $\{(p_{i1}, [\underline{q}_{i1}, \overline{q}_{i1}]), \dots (p_{iM_i}, [\underline{q}_{iM_i}, \overline{q}_{iM_i}])\}$. Each price quantity pair $(p_{ij}, [\underline{q}_{ij}, \overline{q}_{ij}])$, specifies the per-unit price, p_{ij}, that supplier i is willing to provide for marginal items in the interval, $[\underline{q}_{ij}, \overline{q}_{ij}])$. The ranges in the volume discount must be contiguous. Let z_{ij} denote the number of units sourced above $\underline{q}_{ij}$ from supplier i, with $z_{ij} \leq \overline{q}_{ij} - \underline{q}_{ij}$. The total price for quantity $(z_{ij} + \underline{q}_{ij})$ is:

$$p(z_{ij}) = p_{ij}z_{ij} + \sum_{\hat{j}=1}^{j-1} p_{i\hat{j}}(\overline{q}_{i\hat{j}} - \underline{q}_{i\hat{j}})$$

The price schedule incorporates an infinite large number of potential indivisible bids from each of the intervals with an XOR constraint across these possible bids.

Associate a decision variable, $x_{ij} \in \{0,1\}$, with each level j of each price schedule i which takes the value 1 if the number of units sourced to supplier i

is in the interval $[\underline{q}_{ij}, \overline{q}_{ij}]$), and continuous variable z_{ij} that specifies the exact number of units sourced above $\underline{q}_{ij}$ from supplier i. Constraints ensure that $z_{ij} > 0 \Rightarrow x_{ij} > 0$. The winner determination formulation for this problem is:

$$\begin{aligned}
&\min_{x_{ij}, z_{ij}} \sum_{i=1}^{N} \sum_{j=1}^{M_i} p_{ij} z_{ij} + x_{ij} C_{ij} \\
\text{s.t.} \quad & z_{ij} - (\overline{q}_{ij} - \underline{q}_{ij}) x_{ij} \leq 0, \quad \forall i, \forall j \\
& \sum_{j} x_{ij} \leq 1, \quad \forall i \\
& \sum_{i} \sum_{j} (z_{ij} + x_{ij} \underline{q}_{ij}) \geq Q \\
& x_{ij} \in \{0,1\}, z_{ij} \geq 0
\end{aligned}$$

where the coefficient C_{ij} computes the total price for all the items purchased up to and include $\underline{q}_{ij}$:

$$C_{ij} = \sum_{\hat{j}=1}^{j-1} p_{i\hat{j}} (\overline{q}_{i\hat{j}} - \underline{q}_{i\hat{j}})$$

A special case of this formulation where each interval in the schedule is a point interval reduces to the multiple choice knapsack problem which is NP-hard [MT80]. Once again the we need to use a change of variables $y_{ij} = 1 - x_{ij}$ to get the canonical maximization form.

Double Auctions and Exchanges. Double auctions are settings with multiple buyers and sellers. There exist two main institutions for double auctions: (i) the continuous double auction, which clears continuously, and (ii) the clearinghouse or *call* auction, which clears periodically.

For homogeneous items, the continuous double auction maintains a queue of bids from buyers sorted in increasing order of price and a queue of offers from the sellers in decreasing order of price. Whenever the offer price is lower than the bid price the bid and ask are matched and the difference is usually kept by the market maker. This requires maintaining a sorted list of asks and bids which is of $O(N \log N)$ where N is the number of active asks/bids.

In this section we focus on call markets, which are more appropriate when bids and asks are combinatorial and with heterogeneous items. The call markets are different in that bids and asks are cleared periodically. The computational aspects of market clearing depends on the market structure [KDL01]. The aspects of market structure that have an impact on winner determination are as follows:

- *Aggregation*: The role of the market-maker in disassembling and reassembling bundles of items. Possibilities include buy-side aggregation, sell-side aggregation or both. If no aggregation is allowed then each bid can be matched to exactly one ask.

- *Divisibility*: The ability to allocate fractions of items, and the ability to satisfy a fraction of agents' bids and asks. When an agent wants its bid or nothing, then its bid is called *indivisible*.

- *Homogeneous/Heterogeneous Goods*: Homogenous goods imply that all the goods being exchanged are all exactly the same and interchangeable (e.g. an auction for a particular financial stocks). If the goods are differentiated, or heterogeneous, then any given ask can only match with a subset of the bids. An important issue related to heterogenous goods is whether they are *substitutes* or *complements*.

The appropriate level of aggregation will depend on the *physical attributes* of the good; e.g. pieces of steel can be cut but not very easily joined (buy-side aggregation), conversely computer memory chips can be combined but not split (sell-side aggregation). Similarly, goods that have multiple attributes often lead to heterogeneous goods. For example, steel coils may differ in the grade or surface quality. Very often *substituting* a higher quality item for a lower quality item is acceptable, e.g. a bid for 10 units of 1.0GHz processors can be substituted with 10 units of 1.2GHz processors with additional cost. In contrast, in some situations the heterogenous good might *complement* each other and provide greater value as a bundle rather than separately. For example an offer for all legs of an itinerary is valuable than a set of disjointed legs. Note that aggregation *does not* imply that the exchange must take physical possession of goods, trades can still be executed directly between agents.

The winner determination problem can be modeled in its most general form as follows. Consider a set of bids B and a set of asks A. Each bid, $b_i \in B$ is associated with a single type of good, and provides a unit bid price, p_i, and a quantity demanded, q_i. Similarly, associated with each ask, $a_j \in A$, is a unit ask price, p_j, and a quantity offered, q_j, for a single type of good. Bids and asks from multiple bidders are assumed to be connected with additive-or logic, and we do not allow bundle bids. This language is sufficiently expressive with substitutable items.

Let $0 \leq x_{ij} \leq 1$ denote the fraction of the demand q_i from bid b_i allocated to ask a_j. For any given bid b_i we also specify a set of asks $A_i \subseteq A$ to which it can be feasibly matched. Similarly, for each ask a_j we specify the set of bids $B_j \subseteq B$ that constitute a feasible match. These assignment restrictions model the feasibility requirements imposed by the heterogeneity of goods. We will restrict our attention to the objective of maximizing surplus without any loss of generality. A general linear formulation can be written that captures all the different market structures in terms of aggregation, divisibility and differentiation. We do not specify the constraints on the variables x_{ij} since this depends on the structure of the market.

$$\max_{x_{ij}} \sum_{i \in A} \sum_{j \in B_j} (p_i - p_j) q_i x_{ij}$$

$$\text{s.t.} \quad \sum_{i \in B_j} q_i x_{ij} \leq q_j, \quad \forall j \in A \tag{1.9}$$

$$\sum_{j \in A_i} x_{ij} \leq 1, \quad \forall i \in B \tag{1.10}$$

$$0 \leq x_{ij} \leq 1 \quad , \forall i, j \tag{1.11}$$

In the simplest case of homogeneous goods we can drop the assignment restrictions, and set $A_i = A$ and $B_j = B$. Assuming divisibility, then x_{ij} indicates the fraction of the available quantity in bid b_i allocated to ask a_j. The matching problem can be solved by sorting the bids in decreasing order of price and offers in increasing price. The crossover point, p^* is the clearing price and bids with price above p^* and asks below p^* are matched.

With assignment restrictions, for example to capture the case of heterogeneous goods, the optimal matching solution can be solved with an LP as long as bids are both divisible and additive-or. The linear program has a network structure which can be exploited to solve the problem efficiently. Any type of aggregation is allowed without impacting the computational complexity of the problem.

On the other hand, if the bids are indivisible, then we have to introduce an integrality structure. We define the decision variable, $x_{ij} \in \{0, 1\}$, as a binary variable that takes a value 1 if bid b_i is assigned to ask a_j and zero otherwise and replace equation (1.11) with $x_{ij} \in \{0, 1\}$. If we restrict the exchange so as not to allow any aggregation then the winner-determination problem is an assignment problem which can be solved very efficiently in polynomial time [AMO93]. Consider a bipartite graph with asks on one side (the asks are differentiated by price and seller) and the bids on the other. The constraint (1.9) can be replaced with $\sum_{i \in B_j} x_{ij} \leq 1$

Otherwise, for example with aggregation on the sell side, the constraint (1.10) with integrality restricts bids to be assigned to at most one ask and the problem becomes the generalized assignment problem which is known to be NP-hard [MT80]. The reader is referred to Kalagnanam et al [KDL01] for a detailed discussion of these issues.

3.3 Multi-Item Auctions

In this subsection we introduce multi-item auctions where multiple heterogenous items are being bought/sold simultaneously. Note that we already discussed several cases of multi-item auctions above by introducing substitutable heterogeneous goods. An alternative setting (to the double auctions discussed above) allows for more complex bundled bids with all-or-nothing offers for the bundle in cases where the goods are em not substitutable rather they are complementary. Such bundled bids also require a more expressive bidding language that allows for XOR bids.

Following the notation in Section 2.3, let $\mathcal{G} = (1, \ldots, N)$ denote the set of items for sale. The bidders are allowed to specify bundles $S \subseteq \mathcal{G}$ with a single price on the entire bundle. We formulate this problem by introducing a decision variable $x_i(S)$ for each bundle S offered by bidder i. Each bidder provides a bid set $B_i \subseteq 2^{\mathcal{G}}$. Let $p_i(S)$ denote the price offered by agent i for bundle S, and consider bids in an exclusive-or language. For the simple case of a single seller with multiple buyers, the maximization problem can now be written as:

$$\begin{aligned} &\max_{x_i(S)} \sum_{S\in B_i} \sum_i x_i(S) p_i(S) \\ \text{s.t.} \quad &\sum_{S\in B_i} x_i(S) \leq 1, \qquad \forall i \\ \sum_{S\in B_i, S\ni j} &\sum_i x_i(S) \leq 1, \qquad \forall j \\ &x_i(S) \in 0, 1, \qquad \forall i, S \end{aligned}$$

This is a set packing formulation and is NP-hard [RPH98]. There are special cases under which the structure of this problem simplifies and allows for polynomial time solutions. All these special cases arise out of constraints that reduce the constraint matrix to be totally unimodular [dVV02]. A common example is the case where adjacent plots of land are being sold and bidders might want multiple plots but they need to be adjacent. However, in general to get a totally unimodular constraint matrix fairly severe restrictions have to be placed on the bid structure (e.g. only one bid per bidder with “adjacency” constraints) and this restricts the expressiveness of the bidding language.

The problem can be generalized to allow multiple buyers and sellers. We denote the set of buyers with $\mathcal{B}$ and the set of sellers with $\mathcal{S}$. We allow sellers to submit bundles $S \subseteq \mathcal{G}$ with a single price on the entire bundle. We formulate this problem by introducing an additional decision variable $y_i(S)$ for each bundle S offered by seller $i \in \mathcal{S}$. Let $m_i(S)$ denote the asking price by seller

i for bundle S, and consider bids in an exclusive-or language. Now, for the case of multiple sellers with multiple buyers, the maximization problem can be written as:

$$\begin{aligned}
\max_{x_i(S), y_i(S)} \sum_{S \in B_i} \sum_{i \in \mathcal{B} \cup \mathcal{S}} & (x_i(S) p_i(S) - y_i(S) m_i(S)) \\
\text{s.t.} \quad \sum_{S \in B_i} x_i(S) \leq 1, & \qquad \forall i \in \mathcal{B} \\
\sum_{S \in B_i} y_i(S) \leq 1, & \qquad \forall i \in \mathcal{S} \\
\sum_{S \in B_i, S \ni j} \sum_{i} (y_i(S) - x_i(S)) \geq 0, & \qquad \forall j \\
x_i(S) \in 0, 1, & \qquad \forall i, S \\
y_i(S) \in 0, 1, & \qquad \forall i, S
\end{aligned}$$

The single seller problem is a special case of this and hence the complexity of this remains NP-hard. Notice however that if we restrict the matching to allow no aggregation then the problem becomes assignment problem. For each bundle from a supplier we allow exactly one match to a bundle requested by the bidder. Similarly, each bundled bid form a bidder is restricted to match exactly one bundled offer. This reduces to an assignment problem. However, since each agent can bid a power set $S \subseteq \mathcal{G}$ the assignment problem can become exponential in the number of bids.

3.3.1 Multiattribute Auctions.

Multiattribute auctions relate to items that can be differentiated on several non-price attributes such as quality, delivery date etc. In order to evaluate different offers for a item with different attribute levels we need to appeal to multiattribute utility theory to provide a tradeoff across these different attributes. One common approach assumes *preferential independence*, and supposes that an agent's valuation for a bundle of attribute levels is a linear-additive sum across the attributes. Another more general approach captures nonlinear valuations. It is also interesting to consider both *single sourcing*, in which the buyer chooses a single supplier, and *multiple sourcing*, in which there are multiple items to procure and the buyer is willing to consider a solution that aggregates across multiple suppliers.

Let $\mathcal{J}$ denote the set of attributes of an item, with Q_j to denote the domain of attribute j, and $Q = Q_1 \times \ldots \times Q_M$ denote the joint domain, with $M = |\mathcal{J}|$. Consider a reverse auction setting, and write $v^B : Q \rightarrow \mathbb{R}$ and $c_i(q) : \rightarrow \mathbb{R}$ to denote the buyer's valuation function and cost function of seller $i \in \mathcal{I}$.

preferential-independence The valuation, $v^B(q) = \sum_{j \in \mathcal{J}} v_j^B(q_j)$, where $v_j^B : Q_j \rightarrow \mathbb{R}$ is the buyer's valuation for attribute j. Similarly,

$c_i(q) = \sum_{j \in \mathcal{J}} c_{i,j}(q_j)$, is the cost function of supplier i, for attribute cost functions, $c_{i,j}(q_j)$. Consider a bidding language that captures preferential independence. A bid, $(b_{i,1}, \ldots, b_{i,M})$, from supplier, i, defines a price for attributes $q \in Q$ in terms of a linear combination of prices on each attribute, with $b_i(q) = \sum_{j \in \mathcal{J}} b_{i,j}(q_j)$.

non-linear preferences The valuation, $v^B(q) = V(q_1, ..q_M)$, where $v^B : Q \to \mathbb{R}$. A common assumption is to treat price as being linear and write the valuation as a quasi-linear function $v^B(q) = V(q_1, ..q_{M-1}) - p$. Similarly, $c_i(q) = C(q_1, ..q_{M-1}) - p$, is the cost function of supplier i, for attribute cost functions.

In the case of discrete attributes, the valuations can be enumerated for each feature set and then used within a linear formulation as will become apparent in the formulations developed below. For preferential independence, the evaluation can be done at $\sum_M |Q_i|$ where $|Q_i|$ is the number of levels for attribute i. For nonlinear and quasilinear value functions the evaluation has to be done at $Q = Q_1 \times \ldots \times Q_M$ levels and can become very large.

Single Sourcing:. In a single-sourcing setting only a single winning bid is picked to satisfy the demand. The winner-determination problem is

$$\begin{aligned} \max_{x_{ijk}, y_i} & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{k \in K_j} x_{ijk}(v_{jk} - b_{ijk}) \\ \text{s.t.} \quad & \sum_{k \in K_j} x_{ijk} \leq y_i, \quad \forall i, \forall j \\ & \sum_{i \in \mathcal{I}} y_i \leq 1 \\ & x_{ijk}, y_i \in \{0, 1\} \end{aligned}$$

where $k \in K_j$ indexes the level of attribute j, b_{ijk} is the ask price for level k of attribute j from supplier i, and v_{jk} the buyer's reported value. A straightforward method to solve this problem computes the best attribute values for each supplier, and then chooses the best supplier.

configurable offers: A more interesting setting is when the bid structure is more expressive, and in addition to specifying markup prices for attribute levels as in the preferential-independence bidding language, a supplier can provide configuration rules to indicate which combinations of attributes is infeasible. Similarly, promotions to encourage certain attribute levels can be specified as rules. Propositional logic has been used to capture these rules and these rules can be parsed into linear inequalities and added as side constraints to the winner determination formulations.

An interesting aspect of this setting is that even in the simple case of single sourcing with a budget constraint, the identification of the optimal feature set is NP-hard [BK02]. Consider the simplest setting where the buyer attempts to identify the best configurations from a configurable offer from a single supplier, subject to a budget-constraint, B. Identifying the best configuration can be modeled as a variation of the multiple-choice knapsack problem [MT80]. Again, let $x_{jk} = 1$ indicate that level k of attribute j is selected. Let p_b denote the base price, for a base feature set, and μ_{jk} be the markup associated with choosing level k for attribute j. Assuming an separable additive utility function, then the optimal feature set can be identified as:

$$\max_{x_{jk},p} \sum_{j \in \mathcal{J}} \sum_{k \in K_j} v_{jk} x_{jk} - p$$

$$\text{s.t.} \quad \sum_{k \in K_j} x_{jk} = 1, \quad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} \sum_{k \in K_j} \mu_{jk} x_{jk} + p_b \leq p$$

$$p \leq B,$$

$$x_{jk} \in \{0,1\}, \quad \forall j, k$$

Bichler et al[BKL02] provide a detailed discussion of this configurable offers problem with multiple sourcing and other side constraints.

Multiple Sourcing. There are settings where it might be necessary to source to more than one supplier either because none of the suppliers are large enough to satisfy the demand or business rules may requires a minimum number of suppliers. Let Q denote the buyer demand and let q_i denote the supply of seller i. We will use the same notation as for the single sourcing case. If the bids are divisible then identifying the optimal bids is straightforward - the bids are sorted in descending order of value $v_i^B(b_i)$ and the optimal set of bids are picked from this sorted list until $\sum_i q_i = Q$. Notice that the last bid may be chosen fractionally.

However, if the bids are indivisible then the winner determination problem reduces to a knapsack problem and becomes NP-hard. The winner determination problem can be written as follows:

$$\begin{aligned}
&\max_{x_{ijk}, y_i} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{k \in K_j} x_{ijk}(v_{jk} - b_{ijk}) \\
\text{s.t.} \quad & \sum_{k \in K_j} x_{ijk} \leq y_i, \quad \forall i, \forall j \\
& \sum_{i \in \mathcal{I}} q_i y_i = Q \\
& x_{ijk}, y_i \in \{0, 1\}
\end{aligned}$$

In practice it might be more realistic to impose an acceptable range for the demand. An interesting variation that emerges in multiattribute, multi-sourcing setting is the homogeneity constraint that requires that all selected bids have the same value for some attribute (say color). In order to capture such requirement we introduce an indicator variable z_{jk} that takes a value 1 if any bids are chosen at level k for attribute j. Let T_{jk} denote the set of bids at level k for attribute j, then we can capture this requirement as follows:

$$\begin{aligned}
z_{jk} \leq \sum_{i \in T_{jk}} x_{ijk} \leq |T_{jk}| \, z_{jk} \quad & \forall j, k \\
0 \leq \sum_{k} z_{jk} \leq 1 \quad & \forall j, k
\end{aligned}$$

Notice that these constraints have to be applied for each attribute level. The reader is referred to Bichler and Kalagnanam [BK02] for more details.

3.4 Business Rules as Side Constraints

In a real world setting there are several considerations beside cost minimization. These considerations often arise from business practice and/or operational considerations and are specified as a set of constraints that need to be specified while picking a set of winning suppliers. Recent work [DK01], [SSGL01b], [SS01a] [BK02] in this area provides a comprehensive overview of the constraint types that are possible. We discuss some of the main constraint classes here:

Budget Limits on Trades A common constraint that is often placed is a upper limit on the total volume of the transaction with particular supplier. These limits could either be on the total spend or on the total quantity that is sourced to a supplier. These types of constraints are largely motivated (in a procurement setting) by considerations that the dependency on any particular supplier is managed. Similarly, often constraints are placed on the minimum amount or minimum spend on any transaction, i.e. if a supplier is picked for sourcing then the transaction should be of a minimum size. Such constraints reduce the overhead of managing a large number of very small contracts.

Marketshare Constraints Another common consideration especially in situations where the relationships are longterm is to restrict the market share that any supplier is awarded. The motivations are similar to the previous case.

Number of Winning Suppliers An important consideration while choosing winning bids is to make sure that the entire supply is not sourced from too few suppliers, since this creates a high exposure if some of them are not able to deliver on their promise. On the other hand, having too many suppliers creates a high overhead cost in terms of managing a large number of supplier relationships. These considerations introduce constraints on the minimum, S_{min}, and maximum, S_{max}, number of winning suppliers in the solution to the winner determination problem.

Representation Constraints These specify requirements such as at least one minority supplier is included in the set of winners. A generalization is to specify the number of winners that are required from different supplier types.

Homogeneity Constraints Multi-sourcing for multiattribute items requires special consideration when picking winners. A common constraint is to specify that all the winning bids must have the same value for some attribute/s. For example, if chairs are being bought from 3 different suppliers for an auditorium, then it is important that the color for all chairs be the same. Such constraints can be generalized to allow selection of winning bids such that for an attribute of interest all bids have values adjacent to each other.

These requirements can be modeled as side constraints within the formulation for the winner determination problems that we have outlined above. However the specific form of these side constraints depends on the market structure. We will not provide formulations for the side constraints for each of the settings, instead illustrate these in the context of specific setting in Section 4.

The interesting aspect of these constraints is how they impact the computational complexity of the winner determination problem. Two constraint classes are most interesting from this point of view since introducing these constraints into any setting transforms the problem into a computationally hard problem.

- Budget Constraints with integrality requirements for the choice of bids leads to a knapsack type constraints and lead a NP-hard problems.
- Minimum/Maximum number of winning supplier requirements introduce integral counts (for those suppliers who have winning bids versus those who do not) and lead to a set-cover type of constraint that make winner determination NP-hard.

3.5 Indirect Revelation Mechanisms

The revelation principle is very useful in the economic design of mechanisms and intermediaries. It focuses attention on incentive-compatibility and equilibrium implementation, and allows the mechanism design problem to be expressed as a well-formulated optimization problem. The main strength of the revelation principle is its ability to hide implementation and computational issues while performing economic design. In situations in which there exists a computationally-reasonable direct implementation of the mechanism, with good computational properties for both agents and the mechanism infrastructure, this can work very well.

However, in other cases it is necessary to implement an *indirect* implementation of the direct-revelation mechanism. An indirect mechanism does not require that every agent provides complete and exact information about its valuation over all possible outcomes, but instead can allow an agent to reveal preference information as necessary, along the equilibrium path. This is important when the direct-revelation mechanism places unreasonable computational requirements on agents (agents must compute their complete preferences), or unreasonable communication requirements (agents must report their complete preferences), or unreasonable computational requirements on the mechanism infrastructure (the mechanism must implement the allocation rules and payment rules based on revealed types). Indirect mechanisms have an additional benefit of decentralizing the *strategic* computation to agents. Although the strategy space is more complex in an indirect mechanism, the rules that map strategies to outcomes are often simpler than in a direct mechanism.[5]

Examples of indirect mechanisms include *ascending-price auctions*, in which agents submit bids in responds to prices and the auctioneer maintains prices and a provisional allocation, and *commodity exchanges* that post a current clearing price and allow buyers and sellers to submit bids and asks. The English auction, which is an indirect implementation of a Vickrey auction. In a Vickrey auction every agent must determine and reveal its value for the item in equilibrium. In progress is made in the English auction as long as any two

[5] Recall that the revelation principle itself follows from a thought experiment in which the mechanism plays strategies for agents and simulates internally an indirect mechanism.

agents bid at the current price. An agent without the highest value simply needs to leave the auction when the price gets too high, and can compute this strategy with an upper-bound on its value and without an exact value for the item. These ideas were first considered by Parkes et al. [Par99b] and have received recent attention in the economics literature [CJ00]. Section 3.5 describes a general primal-dual methodology that can be useful to design indirect mechanisms, such as extensions of the English auction to combinatorial auction settings.

In this section we describe a general methodology to derive iterative auctions, e.g. ascending- and descending-price auctions, which leverages a fundamental connection between linear programming, competitive equilibrium, and the VCG mechanism.

1 Assume myopic best-response strategies. Formulate a linear program (LP) for the efficient allocation problem. The LP should be integral, such that it computes feasible solutions to the allocation problem, and have appropriate economic content. This economic content requires that the dual formulation computes competitive equilibrium prices that support the efficient allocation, *and* that there is a solution to the dual problem that provides enough information to compute VCG payments.[6]

2 Design a primal-dual algorithm that maintains a feasible primal and dual solution, and terminates with solutions that satisfy complementary-slackness conditions and also satisfy the any additional conditions necessary to compute the VCG payments. The algorithm should not assume complete access to agent valuations, but rather access to a myopic best-response *oracle* that responds with a payoff-maximizing allocation given prices.

This primal-dual methodology has been used to develop auctions for the assignment problem [DGS86], combinatorial auctions [PU00a, PU02], multiattribute auctions [PK02], multi-unit auctions [BdVSV01], and shortest-path problems [BdVSV01]. Myopic best-response by agents to a sequence of prices provides enough information to implement primal-dual algorithms that compute efficient allocations. Terminating with the VCG payments makes the auctions indirect implementations of the VCG mechanism, and provides useful equilibrium properties. Myopic best-response becomes a game-theoretic equilibrium, such that there is no better strategy for an agent whatever the

[6] The agents-are-substitutes is a sufficient condition [BO02], in which the CE prices that maximize the total payoff to agents on one-side of the market support the VCG payments to those agents. But the primal-dual methodology does not require that a single dual solution exists that simultaneously supports the VCG payments to every agent. Instead, it is necessary that the VCG payment to each agent is supported in *some* dual solution. When this condition holds then the universal-CE price condition is sufficient and necessary to be able to compute VCG payments from a dual solution [PU02].

preferences of other agents, so long as the other agents also follow myopic best-response [GS00, PU02]. This *ex post* Nash equilibrium is a useful solution concept because agents can play the equilibrium without any information about the types of the other agents. All that is required is that the other agents are rational, and play equilibrium strategies.

3.5.1 Example: The English Auction. To illustrate the primal-dual methodology, we derive the English auction, which is an efficient and strategyproof auction for the single-item allocation problem. Let v_i denote agent i's value for the item. The efficient allocation problem is:

$$\begin{aligned} & \max_{x_i} \sum_i v_i x_i && [\text{IP}_{\text{single}}] \\ \text{s.t.} \quad & \sum_i x_i \leq 1 \\ & x_i \in \{0,1\} \end{aligned}$$

where $x_i = 1$ if and only if agent i is allocated the item, i.e. the goal is to allocate the item to the agent with the highest value. [LP$_{\text{single}}$] is an integral linear-program formulation with suitable economic properties.

$$\begin{aligned} & \max_{x_i, y} \sum_i v_i x_i && [\text{LP}_{\text{single}}] \\ \text{s.t.} \quad & \sum_i x_i + y \leq 1 \\ & x_i \leq 1, \quad \forall i \\ & x_i, y \geq 0 \end{aligned}$$

Variable, $y \geq 0$, is introduced, with $y = 1$ indicating that the seller decided to make no allocation. The dual formulation, [DLP$_{\text{single}}$], is:

$$\begin{aligned} & \min_{p, \pi_i} p + \sum_i \pi_i && [\text{DLP}_{\text{single}}] \\ \text{s.t.} \quad & \pi_i \geq v_i - p, \quad \forall i \\ & p \geq 0 \\ & p, \pi_i \geq 0 \end{aligned}$$

in which dual variable, $p \geq 0$, represents the price of the item. Given a price, p, the optimal values for π_i are $\pi = \max(0, v_i - p)$, which is the maximal payoff to agent i at the price. The CS conditions are:

$$p > 0 \Rightarrow \sum_i x_i + y = 1 \tag{CS-1}$$

$$\pi_i > 0 \Rightarrow x_i = 1, \quad \forall i \tag{CS-2}$$

$$x_i > 0 \Rightarrow \pi_i = v_i - p, \quad \forall i \tag{CS-3}$$

$$y > 0 \Rightarrow p = 0 \tag{CS-4}$$

In words, if the price is positive then the item must be allocated to an agent by (CS-1) and (CS-4); the price must be less than the value of the winning agent by (CS-3) and feasibility $(\pi \geq 0)$; and the price must be greater than the value of all losing agents, so that $\pi_i = 0$ for those agents (CS-2).

The English auction maintains an ask price on the item, initially equal to zero. In each round an agent can bid at the current price or leave the auction. An agent's myopic best-response (MBR) strategy is to bid while the price is less than its value. As long as two or more agents bid in a round, the ask price is increased by the minimal bid increment, ϵ. An agent is selected from the agents that bid in each round to receive the item in the provisional allocation. The bid from the agent in the provisional allocation is retained in the next round. The auction terminates as soon as only one agent submits a bid. The agent receives the item for its final bid price.

We have just described a primal-dual algorithm. The ask price defines a feasible dual solution, the provisional allocation defines a feasible primal solution. The CS conditions hold when the auction terminates, and the final allocation is an optimal primal solution and efficient. Suppose the provisional allocation assigns the item to agent $\hat{\imath}$. Construct a feasible primal solution with $y = 0$, $x_{\hat{\imath}} = 1$ and $x_i = 0$ for all $i \neq \hat{\imath}$. Given ask price, p_{ask}, consider a feasible dual solution with $p = p_{\text{ask}}$. This is feasible as long as $p_{\text{ask}} \geq 0$, with the optimal dual solution given this price completed with payoffs, $\pi_i = \max(0, v_i - p)$.

Conditions (CS-1,CS-3) and (CS-4) are maintained in each round. Condition (CS-2) holds on termination, because $\pi_i = 0$ for all agents except the winning agent, otherwise another agent would have bid by MBR. The English auction also terminates with a price that implements the Vickrey payment. The optimal dual solution, or CE prices, that maximizes the payoff to the winner across optimal solutions, sets $p = \max_{i \neq i^*} v_i$ where i^* is the agent with the highest value. This is the payment by the winner in the equilibrium of the VCG mechanism in this setting, which is the second-price sealed-bid (Vickrey) auction. The English auction terminates with an ask price, p^*, that satisfies $p^* \geq \max_{i \neq i^*} v_i$ and $p^* - \epsilon < \max_{i \neq i^*}$, and implements the Vickrey outcome as $\epsilon \to 0$. In this simple auction this is sufficient to make MBR a *dominant* strategy for an agent, but in more general settings, such as combinatorial auctions, it is sufficient to make MBR an *ex post* Nash equilibrium [PU02].

3.6 Interactions between Computation and Incentives

Limited computational resources, both at agents and within the mechanism infrastructure, and limited communication bandwidth, can often necessitate the introduction of explicit approximations and restrictions within mechanism and market designs, or at least careful design to provide good computational properties in addition to good economic properties. Introducing approximations, for example to the allocation rule in a mechanism, can fundamentally change the economic properties of a mechanism. For example, many approximations to the functions $g_{\text{eff}}(\cdot)$ in the Groves mechanism payment and allocation rules break strategyproofness. We focus in this section on *interactions* between computational considerations and incentive considerations in mechanism design. Just as classic mechanism design introduces IC constraints to restrict the space of feasible mechanisms, computational constraints further restrict the space of feasible mechanisms. We divide our discussion into the following areas:

strategic complexity how much computation is required by agents to compute the game-theoretic equilibrium of a mechanism?

communication complexity how much communication is required between agents and the mechanism to implement the outcome of the mechanism?

valuation complexity how much computation is required by agents to compute, or elicit, enough information about their type to be able to compute the game-theoretic equilibrium?

implementation complexity how much computation is required to compute the outcome of a mechanism from agent strategies?

In addition to identifying tractable special cases, for example for a subset of a larger type space, and developing fast algorithms, computational considerations often make it necessary to impose explicit constraints, for example to restrict the expressiveness of a bidding language or to restrict the range of outcomes considered by the mechanism.

3.6.1 Strategic Complexity.

The *strategic complexity* of a is the complexity of the game-theoretic problem facing an agent. Mechanism design uses a rational model of agent behavior, in which agents compute and play equilibrium strategies given information about the mechanism and given beliefs about the preferences, rationality, and beliefs of other agents. But agents must be able to compute equilibrium strategies to play equilibrium strategies, or at the least the mechanism designer must be able to compute equilibrium strategies and provide a certificate to allow agents to verify that strategies are in equilibrium.

Although the general question of how complex it is to construct a Nash equilibrium in a game remains open [Pap01] a number of hardness results have been established for computing equilibria with particular properties [GZ89]. Given this, it is important to consider the strategic complexity of the particular non-cooperative game induced by a mechanism, and for the appropriate solution concept, such as Bayesian-Nash or dominant strategy. We choose to focus on strategic complexity in incentive-compatible DRMs, which are the mechanisms for which issues of strategic complexity have received most attention.

A first approach is to design mechanisms with tractable strategic problems, such as the class of *strategyproof* mechanisms in which truth-revelation is a dominant strategy equilibrium and optimal for every agent irrespective of the types and strategies of other agents. Most work in *algorithmic mechanism design* [NR01] focuses on this class of strategyproof mechanisms and addresses the remaining problems of communication complexity and implementation complexity.

A second approach is to perform mechanism design with respect to explicit assumptions about the computational abilities of agents, such as restricting attention to mechanisms with polynomial-time computable equilibrium. For example, Nisan & Ronen [NR00] introduce the concept of a *feasible best-response*, which restricts the strategies an agent in computing its best-response to a *knowledge set*, which can be a subset of the complete strategy space. Mechanism analysis is performed with respect to a *feasible-dominant* equilibrium, in which there is a dominant-strategy in the restricted strategy space defined by agent knowledge sets. In other work, combinatorial exchange mechanisms (see Section 4.6) are proposed that make small deviations away from truthfulness unbeneficial to agents [PKE01b], and the mechanism design problem has been considered with respect to an ϵ-strategyproofness [Sch01].

It is interesting that limited computational resources can be used as a *positive* tool within mechanism design, for example designing mechanisms in which the only computable equilibria are “good” from the perspective of system-wide design goals. As an example, the problem of strategic manipulation in voting protocols is known to be NP-hard [Bar89], and it is possible to use randomization within a mechanism to make manipulation hard without making the implementation problem for the mechanism hard [CS02a].

3.6.2 Communication Complexity. The *communication complexity* of a mechanism considers the size of messages that must be sent between agents and the mechanism to implement the outcome of a mechanism. To motivate this problem, recall that mechanism design often makes an appeal to the revelation-principle and considers direct mechanisms. However, direct mechanisms require agents to report complete and exact information about their type, which is often unreasonable in problems such as combinatorial auctions. In the worst-case the VCG mechanism for a combinatorial auction requires each agent to submit 2^M numbers, given M items, to report its complete valuation function.

A first approach to address the problem of communication complexity is to implement *indirect mechanisms* (see Section 3.5), which do not require the complete revelation of an agent's type. Instead, an agent must report its strategy to the mechanism *along the equilibrium path*. As an example, whereas the VCG mechanism for a combinatorial auction requires complete revelation of an agent's valuation function, an agent must only provide *best-response* bid information in response to prices in an ascending-price combinatorial auction. Although all mechanisms have the same *worst-case* communication complexity in the combinatorial auction setting [NS02], indirect mechanisms reduce the communication required in many instances of the problem [Par01, chapter 8].

A second approach introduces *compact* representations of agent preferences via the careful design of bidding languages (see Section 3.1). Nisan [Nis00] notes a tradeoff between the *compactness* of a language, which measures the size of messages required to state an agent's preferences, and the *simplicity* of a language, which considers the computation required to evaluate the value of any particular outcome given a message in the language. At one extreme, one could allow agents to submit *valuation programs* [Nis00], that provide the mechanism with a method to compute an agent's value for an outcome on-the-fly, as demanded by the implementation of the mechanism. Valuation programs can be useful when the *method* used to compute an agent's valuation for different outcomes can be described more compactly than an explicit enumeration of value for all possible outcomes. However, in practice, valuation programs require considerable trust, for example that a program is faithfully executed by a mechanism and that valuable and sensitive information is not shared with an agent's competitors.

A third approach is to restrict the expressiveness of a bidding language within a mechanism to provide compactness. In restricting the expressiveness of a bidding language it is important to consider the effect on the equilibrium properties of a mechanism [Ron01]. For example, a VCG-based mechanism in which agents are restricted to bidding on particular bundles can prevent truthful bidding and break strategyproofness. Monderer et al. [MT01] describe necessary and sufficient conditions on the structure of bundles in the language to maintain strategyproofness[7] and an *ex post* no-regret property that states that at termination no agent wants to provide any information about its value that was not already permitted within the language. Related work has considered mechanism design within a class of mechanisms in which severe bounds are imposed on the amount of communication permitted between agents and the mechanism [BN02].

3.6.3 Valuation Complexity. The valuation complexity of a mechanism considers the complexity of the problem facing an agent that must determine its type. There are many settings in which it is *costly* to provide complete and exact information value information, for all possible outcomes. This valuation cost can arise for *computational* reasons [San96], for example in a setting in which an agent's value for a particular procurement outcome is the solution to a hard optimization problem. Consider a logistics example, in which a firm seeks to procure a number of trucks to deliver goods to its customers. The value that the trucks bring to the firm depends on the value of the optimal solution to a truck scheduling problem. This valuation cost can also arise for *informational* reasons, because an agent must elicit preference information from a user to determine the value for a particular outcome [AM02].

Indirect mechanisms provide one approach to address the problem of valuation complexity. Unlike an incentive-compatible DRM, in which an agent must compute and provide complete information about its preferences to the mechanism, an agent can often compute its optimal strategy in an ascending-price auction from approximate information about preferences. Indirect mechanisms allow incremental revelation of preference information through agent bids, with feedback through prices and provisional solutions to guide the valuation computation of agents [Par01, chapter 8]. One can imagine that prices in an ascending-price auction structure a sequence of preference-elicitation queries, such as "what is your best-response to these prices?" When myopic best-response is an equilibrium, and when agents play that equilibrium, then each response from an agent provides additional information about an agent's preferences, refining the space of preferences that are consistent with the agent's strategy.

Experimental results demonstrate the advantages of indirect over direct mechanisms for a model of the valuation problem in which an agent can refine *bounds* on its value for bundles during an auction [Par99b, Par01]. Related work presents experimental analysis to compare the preference-elicitation costs of different schemes to elicit agent preferences in indirect implementations of combinatorial auctions [CS01, HS02]. Recent theoretical results demonstrate

[7] Truth-revelation is defined as a bid in which an agent reports value $v_i(S) = \max_{S' \subseteq S} v_i(S')$ for all bundles S permitted in the language.

the benefits of indirect vs. direct auctions in the equilibrium of a single-item auction, with a simple valuation model and agents that can choose to refine their valuations during the auction [CJ02], and derive necessary and sufficient conditions on information about agent preferences to be able to compute the VCG outcome in a combinatorial auction [Par02].

3.6.4 Implementation Complexity. The implementation complexity of a mechanism considers the complexity of computing the outcome of a mechanism from agent strategies. For example, in a DRM this is the complexity of the problem to compute the outcome from reported agent values. In an indirect mechanism this is the complexity to update the state of the mechanism in response to agent strategies, for example to update the provisional allocation and ask prices in an ascending-price auction. We choose to focus on the issues of implementation complexity in direct mechanisms, which are the mechanisms in which this has received most attention.

One approach is to characterize restrictions on the type space in which the implementation problem is tractable. For example, the winner-determination problem in the VCG mechanism for a combinatorial auction can be solved in polynomial time with particular assumptions about the structure of agent valuations [RPH98, TKDM00, dVV02]. A number of fast algorithms have also been developed to solve the winner-determination problem in combinatorial auctions, even though the problem remains theoretically intractable [SSGL01a, FLBS99, ATY00]. Recent experimental work illustrates the effectiveness of embedding the *structure* of agent valuations within mixed-integer programming formulations of the winner-determination problem [Bou02].

Sometimes it is necessary to impose explicit restrictions and approximations in order to develop a mechanism with reasonable implementation complexity [NR01]. This problem is interesting because introducing approximation algorithms can often change the equilibrium strategies within mechanisms. For example, the strategyproofness of the VCG mechanism relies on the optimality of the allocation rule. Recall that the utility to agent i in the Groves mechanism is:

$$u_i(\theta) = v_i(g(\hat{\theta}_i, \hat{\theta}_{-i}), \theta_i) + \sum_{j \neq i} v_j(g(\hat{\theta}_i, \hat{\theta}_{-i}), \hat{\theta}_j) - h_i(\cdot)$$

where $\hat{\theta}$ are reported types, $g(\cdot)$ is the efficient allocation rule, and $h_i(\cdot)$ is an arbitrary function of the announced types of the other agents. Truth revelation, $\hat{\theta}_i = \theta_i$, maximizes the payoff of agent i, so that the mechanism implements $g(\theta_i, \hat{\theta}_{-i})$, and maximizes the sum of the first two terms. Now, with an approximate solution, $\hat{g}(\cdot)$, in place of $g(\cdot)$, and information about the reported types,

$\hat{\theta}_{-i}$, of the other agents, the agent should announce a type, $\hat{\theta}_i$, to solve

$$\begin{aligned} &\max_{\hat{\theta}_i \in \Theta_i} v_i(k, \theta_i) + \sum_{j \neq i} v_j(k, \hat{\theta}_j) \\ \text{s.t.} \quad & x = \hat{g}(\hat{\theta}_i, \hat{\theta}_{-i}) \end{aligned}$$

The agent chooses its announced type to correct the error in the approximation algorithm, $\hat{g}(\cdot)$, and improve the choice made with respect to its *true* type and the reported types of the other agents.

It is useful to retain strategyproofness, but allow for a tractable approximation to the efficient function, $g(\cdot)$. Nisan & Ronen [NR00] derive necessary and sufficient conditions for VCG-based mechanisms to maintain the useful property of strategyproofness.[8] Let $R(g, \Theta)$ denote the *range* of the allocation algorithm used within a VCG-based mechanism, i.e. $k \in R(g, \Theta) \Leftrightarrow \exists \theta \in \Theta$ s.t. $\hat{g}(\theta) = k$. A VCG mechanism is *maximal-in-range* if the algorithm, $\hat{g}(\cdot)$ satisfies:

$$\hat{g}(\theta) = \max_{k \in R(\hat{g})} \sum_i v_i(k, \theta_i), \quad \forall \theta \in \Theta$$

When this property holds, there is nothing that an agent can do to correct the approximation error, because this would require changing the range of the algorithm.

Nisan & Ronen use this characterization to demonstrate a negative result for the performance of any range-restricted variation on the VCG mechanism. One can show that any truthful and tractable VCG mechanism for the combinatorial auction must have unreasonable worst-case allocative-efficiency, by constructing a set of preferences for which the efficient allocation is outside the range of the mechanism and that all allocations inside have low values. However, this worst-case bad performance may not be very important in practice, especially in a setting in which the range is carefully selected to provide good performance in most instances that occur in practice. From a positive perspective, the sufficiency of maximal-in-range provides a powerful *constructive* method to build truthful mechanisms with tractable implementation problems: choose a range of outcomes; provide agents with a bidding language that is expressive enough to state their preferences across outcomes in the range; and implement an optimal algorithm with respect to the bidding language and the range.

A number of interesting tractable and strategyproof mechanisms have been suggested for problems in which the full VCG mechanism is intractable. Lehmann *et al.* [LOS99] propose a truthful and feasible mechanism for a combinatorial

[8] The condition, *maximal-in-range,* implies the axiom of 1-efficiency introduced by Tennenholtz et al. [TKDM00] in related work.

auction problem with *single-minded* bidders, each with value for one particular bundle of items. The optimal winner-determination problem remains intractable, even in this single-minded setting. Gonen [BGN02] proposes a truthful and feasible mechanism for the multi-unit allocation problem, and without making any assumptions about agent preferences.

Another idea to address implementation complexity *distributes* computation across the agents that participate within a mechanism. For example, consider providing agents with an opportunity to provide better solutions to the winner-determination problem [RPH98, Bre99]. Recent work in theoretical computer science, in the broad area of *distributed algorithmic mechanism design* [FPS01], considers the computational and communication complexity of distributed implementations of mechanisms. Broad research goals include developing appropriate notions of hardness and complexity classes, and designing mechanisms with good distributed computational properties *and* good incentives [FKSS01, FPSS02]. A key challenge when computation is distributed across participants is to make sure that it is incentive-compatible for agents to implement the algorithm truthfully. This extends the consideration of truthful information revelation, present in classic mechanism design, to also require incentives for truthful information processing and computation.

4. Specific Market Mechanisms

In this section we pick a few mechanisms that are interesting, both from a practical point of view and because they illustrate some of the emerging research directions in the design of electronic auctions, markets and intermediaries. Many of the mechanisms are indirect, with agents providing progressive information about their types and information feedback from the mechanism to guide agent strategies. This observation reinforces the importance of indirect mechanisms in practice. Many of the mechanisms are also implementations of VCG mechanisms, or variations on the VCG mechanism, which serves to highlight the continued importance of Groves mechanisms in the design of practical mechanisms.

4.1 Combinatorial Auctions

Combinatorial auctions are characterized by the ability for agents to submit bids on *bundles* of items. This can be important in settings in which items are complements, e.g. “I only want A if I also get B,” because bundle bids allow agents to express explicit contingencies across items. The applications of combinatorial auctions are numerous, including procurement [HRN02], logistics [LOP$^+$00, EK02], and in resource allocation settings [McM94, RSB82]. Let $\mathcal{G}$ denote a set of discrete items and $\mathcal{I}$ denote a set of agents. Each agent has a valuation, $v_i(S) \geq 0$, for bundles $S \subseteq \mathcal{G}$ and quasilinear utility functions. The efficient mechanism design problem has received the most attention, and no general solution is known for the *optimal* (revenue-maximizing) combinatorial auction.

The VCG mechanism provides an efficient *sealed-bid* auction, in which agents submit reported valuation functions in a single-shot auction. Given a suitably expressive bidding language, which allows agent i to describe its valuation, v_i, this is an efficient and strategyproof solution. However it is often unreasonable to expect agents to provide valuations on all possible bundles of items. The valuation problem for a single bundle can often be time-consuming, and more difficult to automate than other processes such as winner-determination and bidding.

Given these objections to one-shot combinatorial auctions there has been considerable interest in the design of *iterative* combinatorial auctions, which can reduce the valuation work required by agents because optimal strategies must only be computed along the equilibrium path of the auction. Proposals for iterative auctions can be described along the following two directions:

bidding language Auctions such as RAD [DKLP98] and AUSM [BLP89] allow participants to submit *additive-or* bids, while other auctions [Par99a, PU00a, GS00, AM02] allow participants to submit *exclusive-or* bids.

information feedback Auctions such as AUSM, the proposed FCC combinatorial auction #31, and the Chicago GSB auction, provide *linear-price* feedback along with the provisional allocation. Auctions such as *i*Bundle and AkBA provide *non-linear price* feedback along with the provisional allocation, and *i*Bundle can introduce dynamic non-anonymous pricing. Other proposals, such as RAD and an ascending-proxy design [AM02] are described without explicit price feedback.

Early theoretical results exist for particular restrictions on agent valuations, including unit-demand preferences [DGS86, Ber88], in which each agent wants at most one item, and *gross-substitutes* preferences [KC82, GS00], which is a technical condition that captures a wider class of preferences but still excludes synergies across items. Recently, an efficient auction, *i*Bundle [Par99a, PU00a], has been developed for *buyer-submodular* preferences[9], which is a slightly weaker condition than gross-substitutes. Myopic best-response (MBR) is a game-theoretic equilibrium in *i*Bundle with buyer-submodular preferences [AM02]. A simple extension, *i*Bundle & Adjust [PU00b, Par01], brings MBR into equilibrium for the slightly wider class of agents-are-substitutes preferences.[10] *i*Bundle Extend & Adjust (*i*BEA) extends the auction for a few ad-

[9] Let $w(L)$ denote the value of the efficient allocation to agents $L \subseteq \mathcal{I}$. Buyer submodular requires $w(L) - w(L \setminus K) \geq \sum_{i \in K} [w(L) - w(L \setminus i)], \forall K \subset L$, and all $L \subseteq \mathcal{I}$.

[10] Agents-are-substitutes preferences require $w(\mathcal{I}) - w(\mathcal{I} \setminus K) \geq \sum_{i \in K} [w(\mathcal{I}) - w(\mathcal{I} \setminus i)], \quad \forall K \subset \mathcal{I}$.

ditional rounds to provide enough information to compute VCG payments, which are implemented at the end of the auction as a discount from the final prices. MBR is an equilibrium of *i*BEA for all agent preferences, even when agents-are-substitutes fails.

Subsequently to *i*Bundle, Ausubel & Milgrom [AM02] have described a proxy-agent variation, in which bidders must submit preferences to proxy agents that submit ascending bids to an auction. The auction implements the outcome in *i*Bundle for reported valuations to the proxy agents. An interesting equilibrium analysis demonstrates the importance of a *bidder-monotonicity property*, which is satisfied by equilibrium outcomes in the ascending-proxy auction but not satisfies by the VCG mechanism. Bidder-monotonicity, which requires that the revenue to the auctioneer must increase when the number of agents increases, provides robustness against joint deviations and shill bidding by agents. However, whenever values are not buyer-submodular there are many equilibrium in ascending-proxy, and agents must solve an implicit bargaining problem to implement equilibrium outcomes.

Many iterative combinatorial auctions [Ber88, DGS86, PU00a, BdVSV01] can be interpreted within the primal-dual design methodology described in Section 3.5. The *i*Bundle auction implements a primal-dual algorithm for a hierarchy of LP formulations for the combinatorial allocation problem [BO02], each of which enriches the price space through the introduction of additional primal variables and constraints.

4.1.1 Case Study: *i*Bundle.

As an illustrative example we describe *i*Bundle(2), which is a simple variation of *i*Bundle in which prices are anonymous, and every agent faces the same price on every bundle. For the purpose of this exposition we will assume MBR, but as discussed above, this is in *equilibrium* strategy with additional assumptions about preferences, and can also be brought into equilibrium through an extension to *i*BEA. The structure for *i*Bundle(2) is a single seller with multiple buyers where the seller is selling multiple items to one or more bidders (multiple sourcing).

*i*Bundle(2) is an ascending-price combinatorial auction, in which agents can bid on arbitrary bundles of items. The auction maintains bundle prices and a provisional allocation, which is computed to maximize revenue in each round, given agent bids. The auction proceeds in rounds, indexed $t \geq 1$. We describe the bids that agents can place and the rules that are used to compute the provisional allocation and increase prices.

In each round an agent can submit exclusive-or bids for bundles, e.g. $(S_1, p^t_{\mathrm{bid},i}(S_1))$ XOR $(S_2, p^t_{\mathrm{bid},i}(S_2))$, to indicate than it wants either all items in S_1 or all items in S_2, but not both S_1 and S_2. The first bidding rule requires that agents resubmit bids for bundles that they are winning in the current provisional allocation.[11] The second bidding rule requires that all bid prices are greater-than or equal to the ask price, except in a couple of special cases, when the agent can take an ϵ-*discount*. Ask prices on bundles are increased across rounds to ϵ above the highest bid from an unsuccessful agent, where $\epsilon > 0$ is the *minimal bid increment*.[12] An agent can repeat its bid at the same price when bidding on a bundle that it is currently winning, even if the ask price has increased.[13] An agent can also bid at ϵ less than the ask price when making a "last-and-final" bid, which states that it will never increase its bid price on any bundles in future rounds.[14]

The auction takes the bids in each round, and solves a winner-determination problem, which computes the provisional allocation that maximizes the revenue given agent bids. This winner-determination problem is formulated as a weighted set-packing problem, subject to additional side-constraints to respect agent XOR bid constraints.[15] The auction terminates whenever every agent that is still submitting a bid at, or above, the ask price receives a bundle in the provisional allocation. Otherwise, prices are increased, the new allocation and prices are provided as feedback to agents, and the auction continues. On termination the provisional allocation becomes the final allocation, and agents pay the final bid prices.

This *i*Bundle(2) variation is efficient with MBR strategies for preferences that induce bids that satisfy a technical condition of *safety*. Define agent i's *personalized* price, $p_i(S)$, as the ask price it faces in a particular round, which can be ϵ below the ask price when the agent takes an ϵ-discount. Given this, MBR requires that agent i submits bids for bundles in this set

$$\mathcal{B}_{\text{MBR},i} = \{S \;:\; v_i(S) - p^i_{\text{ask}}(S) + \epsilon \geq \max_{S'}(v_i(S') - p^i_{\text{ask}}(S'), 0)\}$$

This is the set of bundles that come with ϵ of maximizing its payoff at the current prices. *Safety* requires that whenever the agent is not successful in a particular round, all the bundles on which it submitted a bid at or above the ask price are non-disjoint. Preferences that induce safe bids in a MBR equilibrium include agents with additive or superadditive valuation functions, and agents with "required+" preferences, such that the agent builds bundles with positive value around the same core set of required items [PU00a].

[11] One can strengthen this rule, and require that all bids submitted in previous rounds are maintained in a single XOR bid set without changing the analysis of the equilibrium properties of the auction.

[12] An agent is successful whenever one of it is allocated a bundle in the provisional allocation. In this case its bids on other bundles in its XOR bid set are not used to drive up prices.

[13] As a consequence of the winner-determination rules the current ask price cannot be more than ϵ, the minimal bid increment, greater than the agent's bid price.

[14] This is used as a technical device to implement the efficient allocation that includes bidders with very small equilibrium payoffs. An equivalent implementation simply retains *all* bids from agents across rounds.

[15] These constraints can be handled in the set-packing formulation through the addition of *dummy* items to represent each agent's XOR bid.

Given MBR and bid-safety, *i*Bundle maintains feasible primal and dual solutions to the following LP formulation, and terminates with solutions that satisfy CS conditions. The proof technique is inspired by Bertsekas' [Ber87] analysis of the AUCTION algorithm for the assignment problem.

$$\max_{x_i(S),y(k)} \sum_S \sum_i x_i(S) v_i(S) \qquad [\text{LP}_2]$$

$$\text{s.t.} \quad \sum_S x_i(S) \leq 1, \qquad \forall i \qquad (\text{LP}_2\text{-}1)$$

$$\sum_i x_i(S) \leq \sum_{k \ni S} y(k), \qquad \forall S \qquad (\text{LP}_2\text{-}2)$$

$$\sum_k y(k) \leq 1 \qquad (\text{LP}_2\text{-}3)$$

$$x_i(S), y(k) \geq 0, \qquad \forall i, S, k$$

$$\min_{\pi_i, p(S), \Pi} \sum_i \pi_i + \Pi \qquad [\text{DLP}_2]$$

$$\text{s.t.} \quad \pi_i + p(S) \geq v_i(S), \qquad \forall i, S \qquad (\text{DLP}_2\text{-}1)$$

$$\Pi - \sum_{S \in k} p(S) \geq 0, \qquad \forall k \qquad (\text{DLP}_2\text{-}2)$$

$$\pi_i, p(S), \Pi \geq 0, \qquad \forall i, S$$

This formulation introduces auxiliary variables, $y(k)$, where $k \in K$ is a *partition* of items in set K, and K is the set of feasible partitions, to strengthen the LP relaxation in Section XX [BO02]. A feasible partition defines a feasible "bundling" of items, e.g. $[A, B, C]$ or $[AB, C]$, etc., are feasible partitions of items ABC. Given partition k, we use $k \ni S$ to indicate that S is included in partition k. Constraints (LP$_2$-2) and (LP$_2$-3) replace constraints (LP$_1$-1) xx check this from earlier xx, and ensure that no more than one unit of every item is allocated. The dual [DLP$_2$] introduces variables π_i, $p(S)$ and Π. Variable, $p(S)$, can be interpreted as the *ask price* on bundle S, and with substitution $\pi_i = \max_S\{v_i(S) - p(S), 0\}$, and $\Pi = \max_{k \in K} \sum_{S \in k} p(S)$, the dual objective is to minimize the sum of the maximal payoff to each agent and the maximal revenue to the auctioneer. Optimal dual prices correspond to CE prices whenever the primal LP is integral.

The provisional allocation, $\hat{S}$, and ask prices, $p_{\text{ask}}(S)$, in each round define feasible primal and dual solutions. To construct the primal, set $x_i(\hat{S}_i) = 1$ and $x_i(S') = 0$ for all $S' \neq \hat{S}_i$, and set $y(\hat{k}) = 1$ for $\hat{k} = [\hat{S}_1, \ldots, \hat{S}_N]$, and $y(k) = 0$ otherwise. To construct the dual, set $p(S) = p_{\text{ask}}(S)$, and define

$\pi_i = \{0, \max_S v_i - p(S)\}$ and $\Pi = \max_k \sum_{S \in k} p(S)$.[16] Let $(\hat{P}, \hat{D})$ denote this feasible primal-dual pair. The first primal CS condition is:

$$x_i(S) > 0 \Rightarrow \pi_i + p(S) = v_i(S), \qquad \forall i, S \tag{CS-1}$$

Given the values assigned to π_i in $(\hat{P}, \hat{D})$, this is maintained throughout the auction because a bundle is only allocated to an agent if an agent has bid for that bundle, and agents bid for payoff-maximizing bundles in MBR. Formally, for any bundle S that receives a bid from agent i: (i) $p_{\text{ask}}(S) - \epsilon \leq p_{\text{bid},i}(S) \leq p_{\text{ask}}(S)$; (ii) $v_i(S) - p_{\text{bid},i}(S) + \epsilon \geq \max_{S'}\{v_i(S') - p_{\text{bid},i}(S')\}$ and $v_i(S) - p_{\text{bid},i}(S) \geq 0$ from MBR. We have $x_i(S) > 0 \;\Rightarrow\; v_i(S) - p_{\text{ask}}(S) + 2\epsilon \geq \max\{0,\ \max_{S'}\{v_i(S') - p_{\text{ask}}(S')\}\}$. Substituting for π_i and $p_{\text{ask}}(S) = p(S)$ gives a relaxed formulation of (CS-1):

$$x_i(S) > 0 \Rightarrow \pi_i + p(S) \leq v_i(S) + 2\epsilon, \qquad \forall i, S \tag{ϵ-CS-1}$$

The second primal CS condition is:

$$y(k) > 0 \Rightarrow \Pi - \sum_{S \in k} p(S) = 0, \qquad \forall k \tag{CS-2}$$

This states that the allocation must maximize the auctioneer's revenue at prices $p(S)$, over all possible allocations and irrespective of bids received from agents. Recall that the provisional allocation is selected to maximize revenue *given bids*, so we must show that the restriction to agents' bids comes at no cost. The following conditions hold during the auction:

(P1) All bundles with strict positive prices[17] receive a bid from some agent in every round.[18]

(P2) All bundles in allocations that solve Π actually receive bids from *different* agents.[19]

[16] Notice that it is not necessary to explicitly compute, π_i. But, we use these values to establish the complementary-slackness conditions.

[17] An ask price $p_{ask}(S)$ is strictly positive if the price is greater than the ask price for every bundle contained in S, i.e. $p_{ask}(S) > p_{ask}(S')$ for all $S' \subset S$.

[18] Agent i with one of the highest losing bid for bundle S in round t will continue to bid for bundle S in rounds $t+1$. Let $u_i^t(S)$ denote agent i's payoff for bundle S in round t. Then, $u_i^{t+1}(S) = u_i^t(S) - \epsilon$ because the ask price for S increases by ϵ. Also, $u_i^t(S) \geq u_i^t(S')$ for all bundles S' the agent did not bid in round t. Hence, with $u_i^t(S') \geq u_i^{t+1}(S')$ because the price of S' can only increase in round $t+1$, we have $u_i^{t+1}(S) \geq u_i^{t+1}(S') - \epsilon$ and a bid for S' can never exclude a bid for S from agent i's best-response bids in round $t+1$. A similar argument can be made for bundles that did not receive a bid from the agent in round t.

[19] This follows from the safety property, which prevents a single agent from causing the price to increase to its current level on a pair of disjoint bundles. It is clear that this cannot happen in a single round. Furthermore, it can be shown by induction across rounds that an agent with a myopic best-response bidding strategy cannot increase the price of compatible bundles over a sequence of rounds without submitting unsafe bids in a single round.

From (P1) and (P2), we have:

$$\sum_{S_i \in \hat{k}} p_{\text{bid},i}(S_i) \geq \max_{k \in K} \sum_{S_i \in k} p_{\text{bid},i}(S_i)$$

and together with $p_{\text{ask}}(S) \geq p_{\text{bid},i}(S)$, $p_{\text{bid},i}(S) \geq p_{\text{ask}}(S) - \epsilon$, and the definition of Π, this gives a relaxed formulation of (CS-2),

$$y(k) > 0 \Rightarrow \Pi - \sum_{S \in k} p(S) \leq \min\{M, N\}\epsilon, \qquad \forall k \qquad (\epsilon\text{-CS-2})$$

because an allocation can include no more bundles than there are agents or items. The first two *dual* CS conditions, $p(S) > 0 \Rightarrow \sum_{i \in \mathcal{I}} x_i(S) = \sum_{k \in K, S \in k} y(k), \qquad \forall S$ and $\Pi > 0 \Rightarrow \sum_{k \in K} y(k) = 1$ are trivially satisfied because of the construction of the feasible primal and dual solutions. The important dual CS condition, (CS-3), states:

$$\pi_i > 0 \Rightarrow \sum_{S \subseteq G} x_i(S) = 1, \qquad \forall i \qquad \text{(CS-3)}$$

In words, every agent with positive utility for some bundle at the current prices must receive a bundle in the allocation. This is satisfied in each round for agents that receive bundles in the provisional allocation, and also holds for the unsuccessful agents when the auction terminates because of MBR.

Finally, we can bound the worst-case error of the efficiency of *i*Bundle. First, sum ϵ-CS-1 over all agents in the final allocation, and substitute $\pi_i = 0$ for agents not in the allocation by (CS-3). This gives

$$\sum_{i \in \mathcal{I}} \pi_i \leq \sum_{i \in \mathcal{I}} v_i(\hat{S}_i) - \sum_{i \in I} p(\hat{S}_i) + 2\min\{M, N\}\epsilon$$

because an allocation can include no more bundles than there are items or agents. Then, substitute ϵ-CS-2, because $y(\hat{k}) = 1$ for the partition that corresponds with the final allocation, $\hat{S}$, to give $\Pi \leq \sum_{i \in \mathcal{I}} p(\hat{S}_i) + \min\{M, N\}\epsilon$. Adding these two equations, we have

$$\Pi + \sum_{i \in \mathcal{I}} \pi_i \leq \sum_{i \in \mathcal{I}} v_i(\hat{S}_i) + 3\min\{M, N\}\epsilon$$

The LHS is the value of the final dual solution, $V(\hat{D})$ and the first-term on the RHS is the value of the final primal solution, $V(\hat{P})$. Let V^* denote the value of the optimal primal solution by the weak duality property of linear programs. Thus, because $V(\hat{D}) \leq V(\hat{P}) + 3\min\{M, N\}\epsilon$, then $V(\hat{P}) \geq V^* - 3\min\{M, N\}\epsilon$. As the minimal bid increment, $\epsilon \to 0$, the primal solution converges to the efficient allocation.

4.2 Multi-unit Auctions

In a multi-unit auction there is a set of K indivisible and homogeneous items, and agents have valuations, $v_i(m)$, for $m \geq 0$ units of the item. The efficient auction design problem has received some attention. Looking beyond the direct-revelation VCG mechanism for the problem, the challenge when developing an iterative multi-unit auction is to avoid introducing an explicit identifier for each item, which would reduce the problem immediately to the combinatorial allocation problem. Instead, one would like to allow compact bids, that indicate demand for different *numbers* of units, and at most K prices, with prices for different *numbers* of units.

We will consider the *forward auction* problem, with one seller and multiple buyers, and distinguish between two simple cases. Both cases can both be solved with iterative auctions that maintain a *single* unit price. The first case assumes *unit demand* preferences, in which each agent wants at most one unit. The second case assumes *marginal-decreasing* valuations, and leads to an innovative clinching auction [Aus97]. Both auctions have a direct implementation as primal-dual implementations of an appropriate LP formulation of the allocation problem [BdVSV01].

4.2.1 Unit Demand.

In the unit demand setting, an agent wants a single unit of the good. This has a market structure of a single seller with multiple buyers in a forward auction. The seller offers multiple identical units of a single good and allows bids for a single unit. The available resources are matched to multiple bidders and a single unit price is maintained.

Let v_i denote agent i's value for this unit. The following LP is integral, and the solution to its dual corresponds to a competitive equilibrium price. Let $x_i \geq 0$ denote the number of units assigned to agent i.

$$\begin{aligned} & \max_{x_i} \sum_{i \in \mathcal{I}} x_i v_i \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} x_i \leq K \\ & 0 \leq x_i \leq 1, \quad \forall i \in \mathcal{I} \end{aligned}$$

Introducing dual variables, p and π_i, to correspond with the primal constraints, the dual formulation is:

$$\begin{aligned} & \min_{p, \pi_i} \sum_{i \in \mathcal{I}} \pi_i + Ky \\ \text{s.t.} \quad & p + \pi_i \geq v_i, \quad \forall i \in \mathcal{I} \\ & p, \pi_i \geq 0 \end{aligned}$$

Variable, p, can be interpreted as the price on a unit, and π_i as the maximal payoff to agent i, with optimal values given price p computed as $\pi_i = \max(0, v_i - p)$. CS condition, $p > 0 \Rightarrow \sum_i x_i = K$, requires that the outcome maximizes the seller's payoff at the price and implies that the price must be zero unless all items are sold. CS conditions, $\pi_i > 0 \Rightarrow x_i = 1$ and $x_i > 0 \Rightarrow p + \pi_i = v_i$ imply that agents must receive an item if and only if the item has positive payoff at the price. These are familiar conditions for competitive equilibrium. Moreover, as in the multiattribute auction example, there is a particular dual solution that implements the VCG payoffs. The VCG payoffs are implemented at the *minimal CE price*, which is the smallest p that corresponds to an optimal dual solution.

A simple ascending auction implements a primal-dual algorithm for this problem, terminating in the VCG outcome, and with MBR a dominant strategy equilibrium [BdVSV01]. The auction maintains a single ask price, p_{ask}, and allows agents to bid for a single unit in each round at the ask price. While more bids are received than there are items, K agents are selected in the provisional allocation, and the ask price is increased. The auction terminates as soon as fewer than K bids are received, with items allocated to the agents still bidding and remaining items allocated to agents that were active in the previous round, breaking ties at random.

4.2.2 Marginal-Decreasing Values. The structure of this auction is the same as for the single unit demand. The primary difference is the valuation of the seller is assumed to be *marginal-decreasing*. An efficient ascending-price auction has been proposed for the case of multi-unit items and *marginal-decreasing* valuations [Aus97]. A valuation function, $v_i(m)$, is marginal-decreasing if $v_i(m+2) - v_i(m+1) \geq v_i(m+1) - v_i(m)$, for all $m \geq 0$. The auction maintains a single ask price, but different units of the item are sold along the path of the auction, so that each unit can sell at a different price and agents need not pay the same per-unit price. The auction implements the VCG outcome with MBR strategies, making MBR an *ex post* Nash equilibrium. As we might expect, the auction corresponds with a primal-dual algorithm, and this has been shown for two alternate formulations of the multi-unit allocation problem [BdVSV01, BO00]. Rather than repeating this analysis here, we simply describe the auction and explain its dynamic pricing rule.

The auction maintains a price, p_{ask}, and agents submit bids for a quantity, $q_i(p)$, in each round. The auction terminates as soon as the total quantity demanded is less than or equal to K, and otherwise the price is increased. As the price is increased, and demand drops, agents can "clinch" units of the item. Clinching a unit in round t locks in the price that the agent must pay for that unit to the current ask price. An agent clinches a unit of the item in the first round in which the demand from the other agents is low enough that the agent is sure to win the item at its current bid price, assuming other agents have marginal-decreasing values and follow a MBR strategy.

agent	$v_i(1)$	$v_i(2)$	$v_i(3)$
1	123	236	329
2	75	80	83
3	125	250	299
4	85	150	157
5	45	70	75

Table 1.1. Multi-unit Example. Marginal-decreasing agent values.

Consider the simple example illustrated in Table 1.1, taken from Ausubel [Aus97]. There are 5 agents and 5 units of an item, and agent i has value, $v_i(m)$, for m units. The auction proceeds until $p_{\text{ask}} = 65$, at which time the MBR bids are $q(65) = (3, 1, 2, 1, 0)$, from agents 1,. . .,5 respectively. Let $c_i \geq 0$ denote the number of items clinched by agent i, initially set to zero. Agent 1 clinches the first unit, at the current price, because $c_1 + \sum_{i \neq 1} q_i(p) < K$ $(0 + 1 + 2 + 1 < 5)$. The auction proceeds until $p_{\text{ask}} = 75$, at which time the MBR bids are $q(75) = (3, 0, 2, 1, 0)$. Agent 1 clinches a second unit, at the current price, because $c_1 + \sum_{i \neq 1} q_i(p) < K$ $(1 + 2 + 1 < 5)$. Agent 3 clinches its first unit, at the current price, because $c_3 + \sum_{i \neq 3} q_i(p) < K$ $(0+3+1 < 5)$. The auction proceeds until $p_{\text{ask}} = 85$, at which time the MBR bids are $q(85) = (3, 0, 2, 0, 0)$, and agents 1 and 2 both clinch one more unit each, at the current price. Finally, agent 1 receives 3 units, for total payment $65 + 75 + 85 = 225$, its VCG payment, and agent 3 receives 2 units, for total payment $75 + 85 = 160$, its VCG payment.

4.3 Multiattribute Auctions

Multiattribute auctions [Che93] extend the traditional auction setting to allow negotiation over price and attributes, with the final characteristics of the item, as well as the price, determined dynamically through agents' bids. For example, in a procurement problem, a multiattribute auction can allow different suppliers to compete over both attributes values and price.

We have $\mathcal{I}$ sellers, a single buyer, and $\mathcal{J}$ attributes. Each attribute, $j \in \mathcal{J}$, has a domain of possible attribute values (or *levels*), denoted with abstract set $\mathcal{Q}_j$; for example $\mathcal{Q}_1 = \{$red,yellow,green$\}$ if attribute 1 is the color of an item. The joint domain, across all attributes, is denoted $\mathcal{Q} = Q_1 \times Q_M$. Each seller, $i \in \mathcal{I}$, has a *cost function*, $c_i(q) \geq 0$, for an attribute bundle, $q \in \mathcal{Q}$, and the buyer has a *valuation function*, $v^B(q) \geq 0$. For simplicity, it is useful to assume that $\mathcal{Q}$ contains a null attribute bundle, ϕ, for which $v^B(\phi) = 0$ and $c_i(\phi) = 0$ for all $i \in \mathcal{I}$. The buyers and sellers have quasilinear utility functions.

4.3.1 Efficient Descending Price MultiAttribute Reverse Auction. In this subsection we introduce a reverse auction with a single buyer and multiple suppliers. We assume that a single item is to be bought, however, the decision of which item to buy is influenced both by price and other attributes of the item. This design assumes that the buyer wants to source the demand to a single winner. We focus on the problem with general nonlinear valuation functions, although there is a simpler companion auction for the case of preferential-independence [PK02]. We present an efficient descending-price auction, which maintains nonlinear prices on attribute sets.

The VCG mechanism for the multiattribute allocation problem is efficient but not budget-balanced. Indeed, by the Myerson-Satterthwaite impossibility theorem there is *no* efficient and budget-balanced mechanism, even in Bayesian-Nash equilibrium. As an alternative, one can impose budget-balance and implement a modified VCG mechanism which is budget-balanced but not quite strategyproof, and not quite efficient in equilibrium.

Let $\hat{v}^B(\cdot)$ and $\hat{c}(\cdot)$ denote the reported valuation and cost functions. The modified VCG mechanism implements the trade that maximizes the reported surplus, $\hat{V}(\mathcal{I}) = \max_{q \in \mathcal{Q}, i \in \mathcal{I}} \big(\hat{v}^B(q) - \hat{c}_i(q)\big)$, and the winning seller receives its VCG payment from the buyer. The seller's VCG payment is computed as $\hat{p}_{\mathrm{mvcg}} = \hat{c}_{i^*}(q^*) + (\hat{V}(\mathcal{I}) - \hat{V}(\mathcal{I} \setminus i^*))$, where (i^*, q^*) is the implemented outcome, and $\hat{V}(\mathcal{I} \setminus i^*) = \max_{q \in \mathcal{Q}, i \neq i^*} \big(\hat{v}^B(q) - \hat{c}_i(q)\big)$, the reported value of the second-best outcome, which we denote as $(\tilde{i}, \tilde{q})$.

Truthful bidding is a dominant strategy for sellers in the modified VCG mechanism, and the mechanism is also *ex post* IR for sellers, and *ex post* IR for a buyer for any strategy that (weakly) understates its value. In the full VCG mechanism the buyer simply makes payment $\hat{c}_{\hat{i}}(\hat{q})$. The modified VCG mechanism overcharges the buyer by $\hat{V}(\mathcal{I}) - \hat{V}(\mathcal{I} \setminus \hat{i})$ to achieve budget-balance. In fact, this difference bounds the maximal *ex post* gain that the buyer can hope to achieve in the modified VCG through some non-truthful strategy [Par02]. The timing is *ex post*, in the sense that this we minimize the most that a buyer could gain with perfect hindsight about the bidding strategies of the sellers. In addition, as the auction becomes more competitive and the marginal product of any single buyer becomes negligible then the gains-from-manipulation to the seller tend to zero.

Given the modified VCG mechanism one can use the primal-dual auction design methodology to construct an iterative auction. We illustrate this with a descending-price auction, NONLINEAR&DISCRETE, which maintains prices on bundles of attribute-levels [PK02]. The auction terminates with CE prices that support the outcome of the modified VCG mechanism. This makes MBR an *ex post* Nash equilibrium for the sellers against a class of *consistent* buyer strategies. A *consistent* strategy for the buyer is any strategy that can be represented as a straightforward myopic best-response strategy for some *ex ante* fixed, but perhaps untruthful, cost function.

The auction implements a primal-dual algorithm for the following LP formulation of the multiattribute allocation problem:

$$\max_{x_i(q), x^B(q)} \sum_{q \in \mathcal{Q}} v^B(q) x^B(q) - \sum_{i \in \mathcal{I}} \sum_{q \in \mathcal{Q}} c_i(q) x_i(q) \qquad \text{[MAP-1]}$$

$$\text{s.t.} \quad \sum_{q \in \mathcal{Q}} x_i(q) \leq 1, \quad \forall i \in \mathcal{I} \qquad (1.12)$$

$$\sum_{i \in \mathcal{I}} x_i(q) \geq x^B(q), \quad \forall q \in \mathcal{Q} \qquad (1.13)$$

$$\sum_{q \in \mathcal{Q}} x^B(q) \leq 1 \qquad (1.14)$$

$$x_i(q), x^B(q) \geq 0$$

$$\min_{\pi_i, \pi^B, p(q)} \sum_{i \in \mathcal{I}} \pi_i + \pi^B \qquad \text{[DMAP-1]}$$

$$\text{s.t.} \quad p(q) + \pi^B \geq v^B(q), \quad \forall q \in \mathcal{Q} \qquad (1.15)$$

$$\pi_i - p(q) \geq -c_i(q), \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q} \qquad (1.16)$$

$$\pi_i, \pi^B, p(q) \geq 0$$

Variables $x_i(q) > 0$ imply that seller i provides items with attributes q, and $x^B(q)$ implies that the buyer receives items with attributes q. This formulation has integral extremal solutions, such that $x_i(q), x^B(q) \in \{0, 1\}$ at optimality. Moreover, through the introduction of auxiliary variables to represent the *buyer* (at optimality, $x^B(q) = \sum_i x_i(q)$ for all $q \in \mathcal{Q}$), and the inclusion of redundant constraints (1.12), the dual solution defines CE prices and the formulation is consistent with the primal-dual methodology.

The dual introduces variables, π_i, π^B and $p(q)$, to correspond with constraints (1.12,1.14) and (1.13) respectively, and computes prices on *bundles*, $q \in \mathcal{Q}$, of attributes levels. Given prices, $p(q)$, the optimal dual sets $\pi_i = \max_{q \in \mathcal{Q}} [p(q) - c_i(q), 0]$ and $\pi^B = \max_{q \in \mathcal{Q}} \left[v^B(q) - p(q), 0\right]$. These have the usual interpretation as the maximal payoff to each seller and the buyer.

The CS conditions demonstrate that optimal primal and dual solutions define a competitive equilibrium outcome.[20]

$$\pi_i > 0 \Rightarrow \sum_{q \in \mathcal{Q}} x_i(q) = 1 \tag{CS-1}$$

$$x_i(q) > 0 \Rightarrow \pi_i - p(q) = -c_i(q) \tag{CS-2}$$

$$\pi^B > 0 \Rightarrow \sum_{q \in \mathcal{Q}} x^B(q) = 1 \tag{CS-3}$$

$$x^B(q) > 0 \Rightarrow p(q) + \pi^B = v^B(q) \tag{CS-4}$$

Taken together, (CS-1) and (CS-2) state that the outcome must maximize a seller's payoff at the prices, and (CS-3) and (CS-4) state that the outcome must maximize the buyer's payoff at the prices. There are many prices, $p(q)$, that solve the dual problem. The maximal CE prices, which maximize the payoff of the winning seller, implement the payment in the modified VCG mechanism. Recall that (i^*, q^*) and $(\tilde{i}, \tilde{q})$ are the first- and second-best solutions respectively, given reported costs and valuations. The maximal CE prices set

$$p(q^*) = v^B(q^*) - (v^B(\tilde{q}) - c_{\tilde{i}}(\tilde{q}))$$

with prices on other attribute bundles, $q \neq q^*$, to satisfy condition $v^B(q) - v^B(q^*) + p(q^*) \leq p(q) \leq \min_{i \neq i^*} c_i(q)$. In particular, let $\overline{p}_{\text{ce}}(q)$, denote the maximal CE prices in which the price on attributes, $q \neq q^*$, are maximized, i.e. $\overline{p}_{\text{ce}}(q) = \min_{i \neq i^*} c_i(q)$ for all $q \neq q^*$. The maximal CE prices are characterized by the property that the second-best seller, $\tilde{i}$, is *pivotal*, in that its maximal utility across all attributes is exactly zero. The payment in the modified VCG mechanism is supported at the maximal CE prices. Notice that $\overline{p}_{\text{ce}}(q^*) = v^B(q^*) - (v^B(\tilde{q}) - c_{\tilde{i}}(\tilde{q}))$, which is exactly $c_i(q^*) + V(\mathcal{I}) - V(\mathcal{I} \backslash i^*)$.

Auction NONLINEAR&DISCRETE proceeds in rounds $t \geq 1$, and maintains ask prices, $p^t(q) \geq 0$, on every attribute bundle $q \in \mathcal{Q}$, and a provisional allocation to indicate the current winning seller and attribute bundle. A minimal bid increment, ϵ, determines the rate at which prices are decreased across rounds. The ask prices, $p^1(q)$, are initialized to some value greater than $\min_{i \neq i^*} c_i(q)$. In each round sellers can submit an exclusive-or set of bids, with bid prices less than or equal to the corresponding ask price.[21] The winner-determination problem involves the buyer dynamically. The bids are collected, and the buyer is asked for its preferred bid. This can be partially

[20] We can ignore the condition corresponding with constraints (1.13) because $\sum_i x_i(q) = x^B(q)$ for all q in all optimal solutions.

[21] As in *i*Bundle, a seller can take an ϵ-discount when (a) repeating a bid that is successful in the current provisional allocation, if that price has decreased across rounds; and (b) when making "last-and-final" bids, that will not be reduced in any future round.

automated by the auctioneer as the auction progresses, based on decisions by the buyer in earlier rounds. Alternatively, one can simply ask the buyer for its valuation function at the start of the auction and completely automate winner-determination. The selected bid becomes the provisional allocation and the ask prices on a bundle are decreased to ϵ below the lowest bid price from an unsuccessful seller. The auction terminates as soon the ask prices do not change for two consecutive rounds. The provisional allocation is implemented at the final bid price.

NONLINEAR& DISCRETE terminates with the outcome of the modified VCG outcome when agents follow MBR strategies, as the minimal bid increment, $\epsilon \to 0$. In MBR, sellers submit bids for all bundles that are within ϵ of maximizing their payoff at the current ask prices, and the buyer selects the bid in each round to maximize its payoff, given the bid prices. The auction maintains a feasible primal and dual solution in each round, and conditions (CS-2,CS-3) and (CS-4) always hold. Finally, (CS-1) holds when the auction terminates because all unsuccessful sellers must have non-positive utility at the prices.

In addition, the prices in the auction are (weakly) greater than the maximal CE prices, $\overline{p}_{\text{ce}}(q)$, on all bundles, $q \in \mathcal{Q}$, in every round. The auction terminates with prices on bundles, q^* and $\tilde{q}$, that equal the maximal CE prices on those bundles. Certain incentive properties follow from this iterative implementation of the modified VCG mechanism. Seller MBR forms a game-theoretic equilibrium together with a restricted class of buyer strategies. This class is the set of *ex ante* consistent strategies, in which the buyer commits to a particular reported valuation before the auction begins. One way to enforce this, although losing the advantages of incremental preference elicitation for the buyer, is to require the buyer to submit its preferences to a proxy agent at the start of the auction. MBR is an *ex post* Nash equilibrium for sellers in this buyer-proxied auction.

4.3.2 Optimal Descending Price Multiattribute Reverse Auction. The structure of this mechanism is the same as the efficient mechanism in the previous section except that the mechanism design goal is to maximize the expected payoff of the buyer, which is the difference between her value for the outcome and the price that she pays. An optimal multiattribute auction mechanism is known for a special case of two attributes and continuous attribute levels for same structure as above of a reverse auction for a single good[Che93]. The auction formulates an optimal reservation price to incorporate within the modified VCG mechanism. The auction is one-shot, and truth-revelation remains a dominant strategy for sellers. The effect of the reservation price is to decrease the average payment received by the winning seller, but runs the risk of missing an efficient trade in some instances. The analysis has also been extended to the case of *correlated* seller costs [Bra97].

However, no optimal auction mechanism is known for a more general formulation of the multiattribute allocation problem. A multi-round payoff-maximizing auction procedure is proposed for a setting with a class of parameterized seller cost functions and naive sellers [BW01]. The functional form of the cost functions are assumed, and the problem facing the buyer is to determine the values of L parameters. These parameters define seller cost functions and provide the buyer with enough information to extract maximal surplus from the winning buyer. The auction uses L rounds to estimate the seller costs functions, with each round implementing the outcome of the modified VCG mechanism for a different buyer valuation function. Each round is implemented as a descending-price multiattribute auction. The valuation function is adjusted across rounds to infer enough information about costs from seller bids to exactly determine the unknown cost parameters. The analysis makes the strong assumption that sellers follow MBR in every round of the auction, which is perhaps unlikely given that the buyer is in a powerful position of being able to restart the auction multiple times. Finally, after L rounds, there is enough information to run one final round and maximize the buyer's payoff, extracting maximal surplus from the winning seller by adjusting the final reported valuation to make the second-best seller as competitive as possible with the winning seller.

4.4 Procurement Reverse Auctions

Reverse auctions are now routinely used for enterprise procurement. Simple single item single sourcing auctions such as the first-price sealed bid and English auctions have found use in the procurement of maintainence, repair and operations. Increasingly, more complex formats such as combinatorial and volume discount reverse auctions are being introduced in strategic sourcing decisions for material and services used in the production process of an enterprise.

Davenport et al [HRN02] have studied the use of reverse multi-unit auctions with volume discounts in a procurement setting. The combinatorial auction is used for single units (lot) of multiple items. All-or-nothing bids are allowed and nonlinear prices are used to feedback information. In contrast, volume discount auctions are used when the multiple units of multiple items are being procured. However, it restricts the bids to be specified for each item and allows the specification of volume discounts.

An interesting (and complicating) issue that arises in this setting is that there are various business rules that are used to constrain the choice of winners. These business rules appear as side constraints in the winner determination problem. Another consideration in procurement auctions is that the outcome should be such that the final prices should be profitable for both the buyer and the suppliers, i.e. a win-win outcome. Davenport et al [DK02] have used the competitive equilibrium property to operationalize this notion of achieving a win-win outcome.

The main challenge in analyzing the properties of such procurement reverse auctions is that the allocation problems relevant are integer programs and a direct appeal to primal-dual algorithms cannot be made. If an appropriate extended formulation can be identified that has the integrality property (i.e. solving this problem as an LP yields feasible integral and optimal results) then an appeal can be made once again to primal-dual algorithms to show competitive equilibrium [BO02]. For forward combinatorial auctions (which lead to set packing problems) with no side constraints such extended formulations have been identified and used to design iterative combinatorial auction [PU00a]. These formulations are large and provide dual prices on bundles and hence require price signaling to be done on bundles rather than items.

The reverse combinatorial auction has two significant variations: (i) set covering formulation rather than set packing, and (ii) the inclusion of business rules as side constraints that complicate the formulation. Davenport et al [DK01] have shown that a similar extended formulation can be derived for the reverse combinatorial auction with side constraints. Once again this extended formulation yields dual prices on bundles of items. In this section we provide a set covering formulation for the winner determination problem for both the combinatorial and volume discount curves (using a Dantzig-Wolfe we present this as a set covering problem). We then provide an iterative descending price auction and show that they yield a competitive equilibria by providing extended formulations that have the integrality property.

4.4.1 Winner Determination for Combinatorial Auctions with Side Constraints.

A mixed integer programming formulation for the reverse combinatorial auction with no side constraints can be written as a set covering formulation as follows:

$$\min_{x_i(S)} \sum_{S \in B_i} \sum_i x_i(S) p_i(S)$$

$$\sum_{S \in B_i, S \ni j} \sum_i x_i(S) \geq 1, \qquad \forall j$$

$$x_i(S) \in 0, 1, \qquad \forall i, S$$

for bids in set $B_i \subseteq 2^{\mathcal{G}}$ with prices $p_i(S)$ on $S \in B_i$. The following side constraints are often encountered in practice and can be added to the MIP formulation as side constraints as follows:

Min/Max Winning Suppliers

$$y_i \leq \sum_S x_i(S) \leq K y_i, \quad \forall i \in N$$
$$W_{i,\min} \leq \sum_i y_i \leq W_{i,\min}$$

Min/Max Quantity per Supplier Assuming that the total amount associates with a bid is $Q_i(S)$ and $W_{i,min}$ and $W_{i,max}$ are the minimum and maximum quantities allowed for each supplier

$$W_{i,\min} \leq \sum_S \sum_{j \in S} x_i(S) Q_i(S) \leq W_{i,\max}$$

Reservation Prices If the reservation prices are specified over bundles then

$$x_i(S) p_i(S) \leq r_i(S)$$

where $r_i(S)$ is the reservation price. Alternately, if reservation prices are specified for each item then

$$r_i(S) = \sum_{j \in S} \pi_j, \quad \forall i, S$$
$$x_i(S) p_i(S) \leq r_i(S)$$

AND-OR/XOR Bids A typical XOR constraint restricts the number of chosen bids for an agent to be at most one as follows.

$$\sum_{S \in B_i} x_i(S) \leq 1 \ \forall i$$

for bid-set, B_i.

4.4.2 Set-Covering Formulation for Supply Curve Auctions.

Using a Dantzig-Wolfe type decomposition, winner determination for supply curve auctions can also be written as a set covering problem. The supplier curves are additive separable; that is,

$$p^j(x_1, .., x_K) = \sum_i \omega_i^j p_i^j(x_i)$$

where w_i^j are weights and p_i are individual price curves for the commodities. We also assume that each is a piece-wise linear function.

To formulate the model we introduce the concept of supply patterns. A supply pattern A is a vector of length K specifying the amount supplied from each of the commodities $A = (a_1, a_2, ..., a_k)$. The cost of a supply pattern for a particular supplier is computed as $p^j(A)$. A supply pattern is feasible for a supplier if he is able to sell the given amount from each of the commodities and the supplied amounts meet all procurer requirements for the supplier. The set of feasible supply patterns for supplier j is denoted by S^j. Note that there could be an exponential number of feasible supply patterns for each supplier. In the mathematical model we introduce a decision variable for each feasible supply pattern of each supplier: y^s is a decision variable indicating whether pattern s is selected or not, $s \in \bigcup_j S^j$. The basic constraints of this optimization problem will ensure that the procurer's demand is met and that at most one pattern is chosen for each supplier:

$$\sum_j \sum_{s \in S^j} a_k^s y^s \geq Q_k, \quad \forall k$$
$$\sum_{s \in S^j} y^s \leq 1, \quad \forall j$$

The lower, L_j, and upper limit, U_j, for the total number of accepted suppliers will be imposed by the following constraint (in conjunction with the supplier constraints):

$$L_j \leq \sum_j \sum_{s \in S^j} y^s \leq U_j$$

On the other hand, lower and upper limits on the amount of goods supplied by any particular supplier can be encoded in the patterns. Assume that l_k^j and u_k^j are such limits for a particular supplier and commodity and that L^j and U^j are limits for a supplier across all commodities. Then any feasible pattern $s \in S^j$ for supplier j must satisfy the following constraints:

$$l_j^k \leq a_s^k \leq u_j^k, \quad \forall k$$
$$L^j \leq \sum_k a_k^s \leq U^j$$

The objective function of minimizing the procurer's cost completes the mathematical model:

$$z = \min \sum_j \sum_{s \in S^j} p^j(A^S) y^s$$

4.4.3 Descending Price Auction for Reverse Auction with Side Constraints.

As discussed earlier, in order to use primal-dual methods to design an iterative auction requires an integral formulation. In this subsection we provide an extended formulation for the reverse combinatorial auction with side constraints that has the integrality property.

The extended formulation is defined over the space of all feasible partitions of items with agent assignments Γ_{feas}. For every $\mu \in \Gamma_{\text{feas}}$ we define a variable that takes a value 1 if the partition-assignment μ is chosen. $\nu_j(S)$ is the cost of allocating the bundle S to agent j. The variable $y(S,j)$ takes the value 1 if the bundle S is allocated to agent j. Note that the set Γ_{feas} is chosen such that only feasible partition-assignments are allowed. For example a partition-assignment μ that violates the maximum number of winning suppliers is not considered in Γ_{feas}. The minimization formulation shown below is integral (proof not provided here). Notice that all integrality requirements on $y(s,j)$ have been relaxed.

$$
\begin{array}{lll}
\min & \sum\limits_{S\subseteq M}\sum\limits_{j\in N} v_j(S)y(S,j) & \\
\text{s.t.} & & \\
& y(S,j) \geq \sum\limits_{\mu\supseteq S^j} \delta_\mu & \forall j\in N,\ \forall S\subseteq M \\
& \sum\limits_{S\subseteq M} y^\delta(S,j) \geq 0 & \forall j \in N \\
& \sum\limits_{\mu\in\Gamma_{\text{feas}}} \delta_\mu \geq 1 & \forall \mu
\end{array}
$$

The first constraint ensures that for each bundle and each agent the total allocation is at least as large as the partitions chosen containing the bundle with assignments to agent j. The second constraint ensures that the total allocation to agent j is non-negative. The third constraint required that at least one partition-assignment be chosen to satisfy demand. Now we present the dual of this formulation:

$$
\begin{array}{lll}
\max & \sum\limits_{j\in N} \pi_s & \\
\text{s.t.} & & \\
& p_j(S) + \pi_j \leq v_j(S) & \forall j\in N,\ \forall S\subseteq M \\
& \pi_S \leq \sum\limits_{S^j\in\mu} p_j(S) & \forall \mu \in \Gamma_{\text{feas}}
\end{array}
$$

The dual variable $p_j(S)$ corresponds to the first equation in primal and similarly π_j and π_s correspond to the second and third equations in primal. Now if we choose values for these dual variables as follows:

$$\pi_j = \max\left\{0, \max_{S\subseteq M}(v_j(S) - p_j(S))\right\}$$

Each agent j chooses a bundle that maximizes the his/her profit, and

$$\pi_S = \min_{\mu\in\Gamma_{\text{feas}}} \sum_{S^j\in\mu} p_j(S)$$

the buyer makes allocations to minimize cost of procurement. Notice that these choices of dual correspond to the conditions of competitive equilibria.

Choosing the dual variables in this way satisfies the complementary-slackness conditions.

$$y(S,j) > 0 \Rightarrow \pi_j + p_j(S) = v_j(S) \tag{CS-1}$$

$$\delta_\mu > 0 \Rightarrow \pi_S = \sum_{S^j \in \mu} p_j(S) \tag{CS-2}$$

This implies that in each round the bundle prices along with locally profit maximizing suppliers are at a competitive equilibria.

Now a descending price auction with prices on each bundle can be used to reach the competitive equilibria following a primal-dual type algorithm. Notice that the extended formulation uses a variable for each partition-assignment pair thereby introducing price discrimination across agents.

4.5 Capacity constrained allocation mechanisms

An emergent research direction is the examination of mechanisms for decentralized allocation (for multi-item procurement) in the presence of capacity constraints at the suppliers. We discuss two mechanisms that have been proposed in the literature. Both are reverse auctions with a single buyer and multiple suppliers but differ in (i) bid structure that they support and (ii) the feedback that is provided. Both mechanisms assume that a partial allocation against a bid is acceptable to the bidders.

Gallien and Wein [GW00] propose an iterative mechanism where the suppliers bid the unit costs for each item (strategically) and their capacity constraint (truthfully). The buyer uses this information to find a cost minimizing allocation and provides private feedback regarding potential allocation to each supplier. In addition, a *bid suggestion device* is provided by the intermediary that computes the profit maximizing bid for the supplier assuming that all other bids remain the same and that the supplier is willing to share actual unit production costs with the trusted intermediary. An important rule imposed on bidding behavior is *non-reneging* on the price of *each* item to ensure efficiency of the mechanism. An assumption made about the bidding behavior of suppliers is that they are myopic best responders (MBR), i.e. they bid to optimize profits in the next round based on the information about other bids in the current round. Under these assumptions they provide convergence bounds and an ex ante bound on the procurement cost. They use numerical simulations to show that suppliers are incented to reveal true production costs (to the intermediary) under appropriate penalties for capacity overloading.

An alternate iterative approach has been proposed recently by Dawande et al [DCK02]. They use a similar setting as Gallien and Wein [GW00] but relax two fundamental assumptions: (i) They do not impose a non-reneging rule on the price for each commodity, instead require all new bids from suppliers should decrease the total procurement cost by some decrement, and (ii) they provide an oracle that is able to determine (for each supplier) whether a revised bid satisfies the cost decrement without requiring the revelation of production costs or capacity constraints explicitly. They show that that for each supplier, generating a profit maximizing bid that decreases the procurement cost for the buyer by δ can be done in polynomial time. This implies that in designs where the bids are not common knowledge, the each supplier and the buyer can engage in an "algorithmic conversation" to identify such proposals in polynomial number of steps. In addition, they show that such a mechanism converges to an "equilibrium solution" where all suppliers are at their profit maximizing solution given the cost and the required decrement δ.

A buyer is soliciting bids to buy m items $I_1, ..., I_m$. The quantity required for item I_i is $d_i \in \mathbb{Z}_+$. Now we have the two different mechanisms: (i) *Descending price* where the price on each item is decremented in each round [GW00], and (ii) *Descending cost* where the cost of procurement is decremented in each round [DCK02].

4.5.1 Descending Price Mechanism. Each supplier j provides a bid as a vector of unit prices b$_{ij}$ one for each item i and a total capacity c$_j$. The supplier j also specifies the production technology, i.e. the capacity required α_{ij} to produced a single unit of item i. The assumptions are as follows:

1 The suppliers are myopic best responders who bid b$_{ij}$ to maximize profit in the next round given the current information,

2 Suppliers indicate their capacity limits truthfully, and

3 Suppliers accept any fractional allocation of demand for an item.

We now describe an iterative scheme:

Step 1: *(initialization)* Let $(c_j^0, a_{ij}^0, b_{ij}^0)$, $i = 1, 2, .., m, j = 1, 2, ..., n$ be the initial set of bids submitted by the bidders.

Step 2: *(buyer's problem)* The buyer solves an allocation problem to minimize her cost:

$$\begin{aligned} \min \sum_{j=1}^{n} \sum_{i=1}^{m} & b_{ij}(t) x_{ij} \\ \text{s.t.} \sum_{i=1}^{m} \alpha_{ij} x_{ij} & \leq c_j, \quad \forall i \\ \sum_{j=1}^{n} x_{ij} & \geq d_i, \quad \forall i \\ x_{ij} & \geq 0, \quad \forall (i,j) \end{aligned}$$

Let z^t and $\vec{x}^t$ be the optimum cost and the corresponding solution vector, respectively, for this problem. The buyer now reports the allocation $\vec{x}_j^t$ privately to each bidder j, $j = 1, ..., n$.

Step 3 *(bid suggestion device)* The device takes as input from the supplier the production costs v_{ij} and computes the profit maximizing bid assuming the competitors bids remain unchanged and subject to the price decrement rule.

$$\begin{aligned} \vec{b}_i^*(t+1) = \quad & \sup[\arg\ \max \Pi_i(\vec{w}_i, \vec{v}_i, \vec{b}_{-i}(t))] \\ \text{s.t.} \quad & \\ & 0 \leq \vec{w}_i \leq \vec{b}_i(t) \\ & \vec{w}_i \in (\epsilon N)^m \end{aligned}$$

where $\Pi_i(\vec{w}, \vec{v}_i, \vec{b}_{-i}(t))$ is the payoff function for agent i, and $\vec{w}_i$ is the decision vector.

Step 4 (*exit criterion*) If $\vec{b}(T) = \vec{b}(T-1)$, **STOP**; otherwise $t = t + 1$ and return to Step 2.

Properties at Termination.

1 Let T be the final round and $v_i^{(k)}$ denote the k-th order statistic of (v_{i1}, v_{i2}, ..., v_{in}) and let $P = \{j \in 1,..,n|\vec{x}_i(T) = 0\}$, then $x_{ij} > 0 \Rightarrow b_{ij}(T) \leq v_i^{n-|P|+1} + \epsilon$. The winning bid for each item is bounded by some order statistic of the production costs.

2 The total procurement cost can be bounded ex ante if all suppliers have the same technology (same capacity across suppliers for producing a unit of item j). If p is the maximal load number then the total procurement cost is bounded by $\sum_{i=1}^{m} q_i(v_i^{p+m+1} + \epsilon)$. The load number is defined as the smallest p for which $\sum_{j=1}^{p+1} c_j > \sum_i \alpha_i d_i$.

4.5.2 Descending Cost Mechanism.

Here, since the decrement is imposed on the procurement cost, each bidder $j, j = 1, ..., n$ proposes a bid $(c_j, \vec{a}_j)$ where $\vec{a}_j \geq 0$ is the vector indicating the number of units of each item bidder j will supply and $c_j \geq 0$ is the total cost to supply all the items in $\vec{a}_j$. The allocation problem in each round is as follows:

Step 1: *(initialization)* Let $(c_j^0, \vec{a}_j^0)$, $j = 1, 2, ..., n$ be the initial set of bids submitted by the bidders. Let $A^0 = [\vec{a}_1^0\ \vec{a}_2^0 ...\ \vec{a}_n^0] \in \mathbb{Q}^{m \times n}$ and $\vec{c} = (c_1^0, c_2^0, ..., c_n^0) \in \mathbb{Q}^m$. Set $t = 0$. Note that $\mathbb{Q}$ is the set of rational numbers.

Step 2: *(buyer's problem)* The buyer solves an allocation problem to minimize her cost:

$$\min\{\vec{c}^t \vec{x} : A^t \vec{x} \geq d; 0 \leq \vec{x} \leq 1\}$$

Let z^t and $\vec{x}^t$ be the optimum cost and the corresponding solution vector, respectively, for this problem. The buyer now executes the following steps:

(a) Reports the allocation $\vec{x}_j^t$ privately to each bidder j, $j = 1, ..., n$.

(b) Computes the set S of column vectors, with their corresponding costs, such that any one of these columns when introduced in the constraint matrix A^t guarantees a decrease in the optimum solution to the allocation problem by at least δ.

Step 3 *(algorithmic interaction between buyer and bidders)* The profit of bidder j from the allocation $\vec{x}_j^t$ is $\vec{p}_j^t = (c_j^t - \vec{a}_j^t \vec{v}_j)\vec{x}_j^t$, where $\vec{v}_j \in \mathbb{Q}_+^m$ is the vector of unit production costs for bidder j.

Let $S_0 = \{(c^{'}, \vec{a}^{'}) : \vec{a}^{'} \leq \vec{d}\} \subseteq \Re^{n+1}$. The optimization problem for the bidder j is

$$L^j : \quad \max\{c^{'} - \vec{v}_j \vec{a}^{'} - \vec{p}_j^t : (c^{'}, \vec{a}^{'}) \in S \cap S_0\}$$

The buyer and bidder j, $j = 1, ..., n$, cooperate privately to solve the optimization problem L^j. Note that L_j is solvable in polynomial time [DCK02].

If it exists, let (c_j^*, a_j^*) be the optimum solution vector for L^j. If the optimum value to L^j is positive, bidder j updates her bid: $(c_j^{t+1}, \vec{a}_j^{t+1}) = (c_j^*, \vec{a}_j^*)$. Otherwise, bidder j retains her previous bid $(c_j^{t+1}, \vec{a}_j^{t+1}) = (c_j^t, \vec{a}_j^t)$.

Step 4 (*exit criterion*) If $(c_j^{t+1}, \vec{a}_j^{t+1}) = (c_j^t, \vec{a}_j^t)\ \forall\, j$, **STOP**; otherwise $t = t+1$ and return to Step 2.

Properties at Termination. Given $\delta > 0$, the allocation problem satisfies the following properties at termination.

- Given the bids, the buyer minimizes her cost.
- For each bidder j, given the bids of all other bidders $k \neq j$, $\nexists$ any bid $(c^{'}, \vec{a}^{'})$, which if were to propose would satisfy the following two conditions simultaneously: (i) gives more profit and (ii) reduces the buyers cost by at least δ.

An interesting *ex-post* property when the iterative scheme terminates is:

- At termination, let the buyer's total cost be c^{*}. Then $\exists\, \delta > 0$ such that, given the bids of all other bidders $k \neq j$, $\nexists$ any bid $(c^{'}\vec{a}^{'}, \bar{c})$ for bidder j which keeps the buyer's cost at c^{*} and increases her profit by more than δ. In words, there exists a value of δ which achieves $\delta-$optimality for each bidder.

4.6 Double Auctions/Exchanges

A key aspect of double auctions is that both the buyers and sellers have private information about their preference structure. This is in stark contract to all the analysis performed so far where either the buyer or the seller was treated as having no reservation price and consideration of incentive compatibility was restricted to a single side of the market. In this section we discuss double auctions and a resulting consideration that arises in this setting for designing incentive compatible mechanisms. The Myerson-Satterthwaite impossibility theorem shows that no efficient (EFF) mechanism can be budget-balanced (BB), with individual-rationality. We discuss double auctions, which are multi-unit homogeneous item allocation problems, and combinatorial exchanges, which are two-sided combinatorial allocation problems.

4.6.1 Double Auctions. The clearing and payment problem can be analyzed as follows. Assume that bids are sorted in descending order, such that $B_1 \geq B_2 \geq \ldots \geq B_n$, while asks are sorted in ascending order, with $A_1 \leq A_2 \leq \ldots \leq A_m$. The efficient trade is to accept the first $l \geq 0$ bids and asks, where l is the maximal index for which $B_l \geq A_l$. The problem is to determine which trade is implemented, and agent payments. In a full VCG mechanism for the double auction, the successful buyers make payment $\max(A_l, B_{l+1})$ and the successful sellers receive payment $\min(A_{l+1}, B_l)$. In general the payments are not budget-balanced, for example with $A_{l+1} < B_l$ and $B_{l+1} > A_l$ and $A_{l+1} > B_{l+1}$.

Table 1.2 surveys some of the double auction mechanisms known in the literature. Only the VCG mechanism is EFF, but it fails BB. All the other mechanisms are strategyproof and BB, except for k-DA, in which truth-revelation is

Name	traded	p_{buy}	p_{ask}	(EFF)	(BB)	(IR)	equil	(IC)
VCG-DA	l	$\max(A_l, B_{l+1})$	$\min(A_{l+1}, B_l)$	Yes	No	Yes	dom	yes
k-DA [CS83, Wil85]	l	$kA_l + (1-k)B_l$	$kA_l + (1-k)B_l$	No	Yes	Yes	Nash	no
TR-DA [BN01]	$l-1$	B_l	A_l	No	Yes	Yes	dom	yes
McAfee-DA [McA92]	l or $l-1$	$(A_{l+1} + B_{l+1})/2$ or B_l	$(A_{l+1} + B_{l+1})/2$ or A_l	No	Yes	Yes	dom	yes

Table 1.2. Double auction mechanisms. The *traded* column indicates the number of trades executed, where l is the efficient number. The *equil* column indicates whether the mechanism implements a dominant strategy or (Bayesian)-Nash equilibrium.

not an equilibrium strategy.[22] In the k-DA [Wil85, CS83, SW89], parameter $k \in [0, 1]$ is chosen before the auction begins; the parameter is used to calculate a clearing price somewhere between A_l and B_l as $kA_l + (1-k)B_l$. The McAfee DA [McA92] computes price $p^* = (A_{l+1} + B_{l+1})/2$, and implements this price if $p^* \in [A_l, B_l]$ and trades l units, otherwise $l-1$ units are traded for price B_l to buyers and A_l to sellers.[23] Mechanisms for exchanges that are (BB) and (IR) fall into two categories (a), that are (IC) and deliberately clear the exchange to implement an inefficient trade, and kind (b), that are not (IC) and clear the exchange to implement the revealed-surplus maximizing trade. Mechanisms TR-DA and McAfee-DA fall into kind (a), while mechanism k-DA falls into kind (b).

4.6.2 Combinatorial Exchanges.

Parkes et al [PKE01b] have suggested a family of VCG-based exchanges in which the exchange is cleared to implement the trade that maximizes reported value (or surplus). The pricing problem is formulated as an LP, to constructs payments that minimize the distance to VCG payments for some metric, subject to IR and BB constraints. A number of possible distance functions are proposed, which lead to simple budget-balanced payment schemes. The authors derive some theoretical properties that hold for the rules, and present experimental results.

The pricing problem is to use the available surplus, V^*, computed at value-maximizing trade λ^*, to allocate discounts to agents that have good incentive properties while ensuring (IR) and (BB). Let V^* denote the available sur-

[22] Recently, Yoon [Yoo01] has proposed a modified version of the VCG-DA in which participants are charged a fee to enter the auction and balance the budget-loss of the VCG payments. Yoon characterizes conditions on agents' preferences under which the modified VCG-DA is (EFF), (IR) and (BB).

[23] Babaioff & Nisan [BN01] have recently proposed the TR-DA rule, which implements the fall-back option of McAfee's DA. The authors also propose an α-reduction DA, in which a parameter $\alpha \in [0, 1]$ is selected before the auction begins. The TR-DA rule is used with probability α, and the VCG DA rule is used with probability $1-\alpha$. Parameter α can be chosen to make the expected revenue zero (and achieve *ex ante* BB) with distributional information about agent values, to balance the expected surplus loss in the VCG-DA with expected gain in the TR-DA. The α-reduction DA is BB, and retains strategyproofness.

plus when the exchange clears, before any discounts; let $I^* \subseteq \mathcal{I}$ denote the set of agents that trade. The pricing problem is to choose discounts, $\mathbf{\Delta} = (\Delta_1, \ldots, \Delta_I)$, to minimize the distance $\mathbf{L}(\mathbf{\Delta}, \mathbf{\Delta}_{\text{vick}})$ to Vickrey discounts, for a suitable distance function $\mathbf{L}$.

$$\min_{\mathbf{\Delta}} \quad \mathbf{L}(\mathbf{\Delta}, \mathbf{\Delta}_{\text{vick}}) \qquad \text{[PP]}$$

$$\text{s.t.} \quad \sum_{i \in I^*} \Delta_i \ \leq V^* \qquad \text{(BB')}$$

$$\Delta_i \leq \Delta_{\text{vick},i} \quad , \forall\, i \in I^* \qquad \text{(VD)}$$

$$\Delta_i \geq 0 \quad , \forall\, i \in I^* \qquad \text{(IR')}$$

Notice that the discounts are *per-agent*, not per bid or ask, and therefore apply to a wide range of bidding languages. Each agent may submit multiple buys and sells, depending on its bids and asks and on the bids and asks of other agents. Constraints (BB') provide worst-case (or ex post) budget-balance, and can be strengthened to allow the market-maker to take a sliver of the surplus (or inject some money into the exchange). One can also substitute an expected surplus $\overline{V}^*$ for V^* and implement average-case budget-balance.

The (IR') constraints ensure that truthful bids and asks are (ex post) individual-rational for an agent, such that an agent has non-negative utility for participation whatever the bids and asks received by the exchange. Constraints (VD) ensure that no agent receives more than its Vickrey discount. The authors consider a variety of distance functions, including standard metrics such as $\mathbf{L}_2(\mathbf{\Delta}, \mathbf{\Delta}_{\text{vick}}) = \sum_i (\Delta_{\text{vick},i} - \Delta_i)^2$ and $\mathbf{L}_\infty(\mathbf{\Delta}, \mathbf{\Delta}_{\text{vick}}) = \max_i (\Delta_{\text{vick},i} - \Delta_i)$. The $\mathbf{L}_1$ metric is not interesting, providing no distributional information because any complete allocation of surplus is optimal. Each distance function leads to a simple parameterized *payment rule*. The payment rules are presented in Table 1.3.

Each payment rule is parameterized, for example the Threshold rule, $\Delta_i^*(C_t) = \max(0, \Delta_{\text{vick},i} - C_t)$, which corresponds to L_2 and L_∞ requires a "threshold parameter", C_t. The final column in Table 1.3 summarizes the method to select the optimal parameterization for each rule. For example, the optimal Threshold parameter, C_t^*, is selected as the smallest C_t for which the solution satisfies BB'. The optimal parameter for any particular rule is typically not the optimal parameter for another rule.

Based on analytic and experimental results, a partial ordering $\{$Large, Threshold $\} \succ$ Fractional $\succ$ Reverse $\succ \{$Equal, Small $\}$ is derived, with respect to the allocative-efficiency of the rules. Although *Large* generates slightly less manipulation and higher allocative efficiency than Threshold in the experimental tests, the Threshold discounts are quite well correlated with the Vickrey discounts while the Large discounts are quite uncorrelated with the Vickrey discounts. This points to the fact that an agent's discount in Large is very sen-

Distance	Name	Definition	Parameter
Function			Selection
$\mathbf{L}_2, \mathbf{L}_\infty$	Threshold	$\max(0, \Delta_{\mathrm{vick},i} - C_t^*)$	$\min C_t$ s.t. (BB')
$\mathbf{L}_{\mathrm{RE}}$	Small	$\Delta_{\mathrm{vick},i}$, if $\Delta_{\mathrm{vick},i} < C_s^*$ 0 otherwise	$\max C_s$ s.t. (BB')
$\mathbf{L}_{\mathrm{RE2}}$	Fractional	$\mu^* \Delta_{\mathrm{vick},i}$	$\mu^* = V^* / \sum_i \Delta_{\mathrm{vick},i}$
$\mathbf{L}_{\mathrm{WE}}$	Large	$\Delta_{\mathrm{vick},i}$, if $\Delta_{\mathrm{vick},i} > C_l^*$ 0 otherwise	$\min C_l$ s.t. (BB')
$\mathbf{L}_{\Pi}$	Reverse	$\min(\Delta_{\mathrm{vick},i}, C_r^*)$	$\max C_r$ s.t. (BB')
-	No-Discount	0	-
-	Equal	$V^* / \lvert I^* \rvert$	-

Table 1.3. Distance Functions, Payment Rules, and optimal parameter selection methods. Constraint (BB') states that $\sum_i \Delta_i^* \leq V^*$, and $\lvert I^* \rvert$ (used in the Equal rule) is the number of agents that participate in the trade.

sitive to its bid, and suggest *Large* is likely to be less robust than Threshold in practice. A complete Bayes-Nash equilibrium analysis is not attempted.

5. Towards Automated Mechanism Design and Evaluation

The standard approach to mechanism design first makes assumptions about the behavior of agents, and about the information available to agents, and then formulates the design problem as an analytic optimization problem to select the optimal mechanism subject to these assumptions. From the revelation principle, mechanism design reduces to optimization over the space of incentive-compatible direct mechanisms. Mechanism design theory is a powerful tool which has produced some very interesting results, both positive and negative. However, mechanism design can fail for any of the following reasons:

problem difficulty The analysis problem can be too difficult to solve analytically. Open problems include the optimal (revenue-maximizing) combinatorial auction, and the most efficient combinatorial exchange amongst the class of budget-balanced exchanges.

inadequacy of direct mechanisms Direct mechanisms are not practical in many settings. Moreover, although primal-dual methods are useful to construct indirect implementations of VCG-based mechanisms, there are no general methodologies to develop indirect implementations of direct mechanisms.

ignorance of computational considerations The analytic approach ignores the strategic, valuation, communication, and implementation complexity of mechanisms. For example, perfect rationality assumptions are implicit within incentive-compatibility constraints (see Section 2.2).

We propose an alternative methodological approach to the design of mechanisms, that we refer to as *automated mechanism design*. The methodology remains within the spirit of classic mechanism design, because we will seek to maximize performance with respect to beliefs about the way that self-interested agents will participate. The basic idea is to *compute* the performance of a particular mechanism with a method to simulate agent strategies, and use this computational method as a “black box” evaluation function against which to perform mechanism design. Rather than compute analytic solutions to the mechanism design problem, mechanism design is performed within a structured search space, expressive enough to capture a class of interesting mechanisms.[24] In addition, since mechanism evaluation is via direct evaluation it is not necessary to limit attention to direct mechanisms for reasons of analytic tractability. Computational considerations can be handled within the model because computational methods are used to design and evaluate the mechanism. For example, automated mechanism design need not be restricted to models of agent behavior that can be captured with analytic expressions, such as incentive-compatibility, but can be performed with respect to strategies, perhaps satisficing and suboptimal, *computed* by a system of agents. In addition, moving away from analytic solutions can capture richer behaviors, such as joint deviations and collusion, during design.

The automated mechanism design approach presents three main challenges, and although each problem has received some attention in isolation in recent years, there certainly remain significant computational difficulties.

evaluation Implement the black box evaluation module, which takes a mechanism description and computes the performance of the mechanism with respect to beliefs about agent behaviors.

optimization Implement the mechanism optimizer, which searches in the space of the mechanism description language for a good mechanism.

description Define a mechanism description language, which is the interface between the mechanism optimizer and the black box mechanism evaluator.

Before we provide a brief review of relevant work in each of these areas, it is worthwhile to note the relationship between to *experimental economics*.

[24] Conitzer & Sandholm [CS02b] use the term automated mechanism design in a different way, to mean the computational optimization of an incentive-compatible based formulation of the mechanism design problem.

Many economists recognize the importance of experimental methodologies to test theoretical predictions, and to assess the robustness of a mechanism to unmodeled behaviors [Mil02, chapter 1],[Smi62, Smi82]. Experiments with human subjects are run in carefully controlled laboratory settings, for example to test predictions about the efficiency of a double-auction market [DH93, chapter4], to compare different combinatorial auction mechanisms [BLP89], and in bargaining settings [RM82]. Indeed, Al Roth, an economist involved in the design of real-world markets, such as those used in the medical resident matching program, advocates an "economics as engineering" approach [Rot02, Var02] to market design.

It has been argued that *agent-mediated* electronic markets, with computational agents instead of human agents, are better suited to theoretical models because agents can be programmed with rational behaviors [Var95]. But still, many equilibrium concepts are hard to compute, and it is often necessary to consider computational constraints explicitly during design. Although strategyproof mechanisms provide one compelling class of mechanisms with computable equilibria, strong impossibility results restrict the applicability of strategyproof mechanisms.[25] Just as economists have turned to laboratory experiments with people to test the predictions of economic theory, it is natural, given our interest in building mechanisms that will be populated by computational agents, that we should turn to computational methods to test, and *design*, mechanisms.

5.1 Automated Mechanism Design: Evaluation

First, given a mechanism we would like to *measure* its empirical performance in a setting that best approximates the actual environment in which it will be used. A direct approach to compute the performance of a mechanism works for game-theoretic agents, and for small problem instances. For example, the GAMBIT toolset[26] is a software program and set of libraries for the construction and analysis of finite games, including the games with incomplete, imperfect information which are important in mechanism design. However, GAMBIT cannot solve even very small problems from sequential auctions in a reasonable amount of time. Zhu & Wurman [ZW02] provide an example in which a five agent, four item auction has 1.5 billion decision nodes. Recent progress has demonstrated that problem representations which capture the explicit structure within a problem can enable speed-ups [KLS01, KM01, KP97].

[25] For example, the Gibbard-Satterthwaite impossibility theorem [Gib73, Sat75] states that it is impossible, in a sufficiently rich environment, to implement a non-dictatorial social-choice function in a dominant strategy equilibrium. A social choice function is dictatorial if one (or more) agents always receives one of its most preferred alternative.

[26] http://www.hss.caltech.edu/gambit/

However, no mechanisms have yet been proposed that induce games with useful computational structure. Moreover, the techniques do not handle alternative solution concepts, such games with computationally-bounded agents.

One alternative to the direct method is to compute a *restricted* equilibrium across a limited set of agent strategies. This approach has been used to analyze strategic interactions in various settings, including continuous double auctions [TD01, DHKT01, Cli97], e-commerce settings with dynamic pricing [KG02], and simultaneous ascending-price auctions [RSCL02]. Even in this setting, with severely restricted agent strategies, it can still prove intractable to compute equilibrium solutions. Wellman et al. [WGSW02] discuss the problem of computing restricted equilibrium in a single-machine scheduling problem with jobs that require multiple time-slots and have deadlines. GAMBIT would often fail with small problems, with five agents, a machine with 5 slots, and 20 strategies. Alternatives, such as an *evolutionary computation* method in which population replicator dynamics [Fri91] were able to successfully converge to a Nash equilibrium, although it was not possible to verify that the solutions were unique. The main shortcoming of the restricted equilibrium approaches is that the initial selection of interesting strategies is quite *ad hoc*, and at the least requires a good deal of insight into the problem.

Another alternative to the direct method is to compute an approximation to the equilibrium of the game. As an example, Zhu & Wurman [ZW02] propose a *fictious play* approach, in which agents sample distributions of types for other agents, and repeatedly compute the Nash equilibrium based on each sample. The equilibrium solutions across all sample games are combined into a single overall equilibrium strategy. In a problem with five agents, that each want a single item, and a sequence of four first-price sealed-bid auctions, experimental results suggest that the performance of the learned strategy is quite close to the optimal equilibrium strategy. Although Zhu & Wurman use this approach without first restricting the strategy space, and are able to adopt GAMBIT to solve any one sample game, this method could also be applied in combination with the restricted equilibrium approaches.

Another alternative to the direct method is to use genetic programming [Koz92] to evolve agent strategies within a mechanism [Pri97, PMPS02]. Phelps et al. [PMPS02] model a continuous double auction, and propose a set of genetic programming primitives from which to evolve trading strategies. The authors demonstrate evolution towards higher efficiency, and are able to provide quite simple and intuitive formulas for the learned strategies. One advantage of this approach, that is shared with the evolutionary computation [WWW02], is that side-steps the issue of computing explicit equilibrium solutions. Another advantage is that it appears to require less *ad hoc* seeding of interesting agent strategies than the restricted equilibrium approaches. However, there are no theoretical characterizations of the solutions computed via genetic programming, and no guarantees that the system converges to a Nash equilibrium.

Finally, another approach is to use a *competition* to test the performance of a mechanism, in which different groups are encouraged to design agents with useful strategies. As an example, the *Trading Agent Competition* (TAC) [WWO$^+$01] has constructed scenarios, and markets, in which automated trading agents compete to win items and maximize their individual utility.[27] TAC has been instrumental in focusing the attention of multiple researchers, and has led to the development of a variety of interesting and novel trading strategies [WGSW02]. The existence of simulators for real-world auction designs, such as the FCC spectrum auction [CLSS01], may make it convenient to run future TAC contests in more realistic domains. In an earlier study, the Sante Fe Institute conducted a tournament for automated trading agents within a double auction [RMP92, RMP94].

5.2 Automated Mechanism Design: Optimization

The *optimization* component of the automated mechanism design challenge problem has received the least attention. One approach is to stick very close to the traditional mechanism design approach, and formulate an optimization problem for a model of game-theoretic agents, with the design space constrained to incentive-compatible and direct mechanisms. Conitzer & Sandholm [CS02b] assume a *discrete* type space and introduce variables to explicitly represent the functions that map reported types to outcomes within the mechanism. Reductions from INDEPENDENT-SET and KNAPSACK [GJ79] show that the mechanism design problem can be NP-hard for deterministic mechanisms, although it is polynomial-time solvable for special cases such as social-welfare maximizing objectives with side payments (by the existence of the Groves mechanisms) and polynomial-time solvable via a linear-program representation for a class of randomized mechanisms.[28] However, even if this direct optimization technique does prove tractable in some special cases, it does not handle alternative solution concepts such as games with computationally-bounded agents, and does not appear well-suited to the optimization of indirect mechanisms.

An alternative approach, suggested in preliminary work by Phelps et al. [PMPS02], is closer to the methodological approach that we outline in this section for automated mechanism design. The authors allow the *genetic pro-*

[27] Many technical reports, describing the design of trading agent strategies, are provided at this URL: http://auction2.eecs.umich.edu/researchreport.html

[28] One implication of their analysis, unless P=NP, is that randomized mechanisms must have greater implementation power for hard deterministic mechanism design problems, unless P=NP.

gramming to evolve the rules of a continuous double auction, with genetic programming primitives that allow the construction of both uniform-price and discriminatory-price rules, including the family of k-DA rules. Experiments were performed in a *co-evolutionary setting*, in which agent rules are also evolving. The results compared the performance of the market with and without auction evolution, and demonstrated a mean efficiency for the adaptive auctioneer of 94.5%, in comparison with a mean efficiency of 74.3% for the static auctioneer. In environments with at least as many buyers as sellers the auction rules converged towards a uniform-price k-DA rule, with k adjusted closer to zero (and the clearing price closer to the minimal successful bid price) as the number of bidders increased. Cliff [Cli01] has also considered the role of genetic algorithms to adapt rules a parameter of the rule set within a double-auction.

5.3 Automated Mechanism Design: Description

Wurman and colleagues [WWW01, WWW02] have developed a parameterization of the auction design space, which is designed to enable the implementation of configurable marketplaces. The features of auctions are characterized according to the methods with which they *handle bids*, *compute outcomes*, and *generate intermediate information*. The Michigan AuctionBot [WWW98], a general-purpose configurable auction server, had provided a catalyst for these efforts. In related work, Reeves et al. [RWG01] have proposed ContractBot, a system that employs a declarative specification of knowledge about alternative negotiation structures and auction rules, and is able to translates a specific negotiation structure into an operational specification for an auction platform.

Perhaps most progress has been made in ontologies for automated negotiation [TWD02, LWJ00], in which an ontology is developed to define the rules of the negotiation between participants within an open system. The motivation for the work in a negotiation ontology is to develop a rich enough language of protocols to provide agents with the ability to *choose* a particular negotiation protocol that is well-suited to their requirements. Tamma et al [TWD02] present a worked example of an application of such an ontology to the Trading Agent Competition.

References

[AMO93] K Ahuja, T L Magnanti, and J B Orlin. *Network flows*. Prentice-Halls, Englewood Cliffs, 1993.

[ATY00] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for auctions with bids for combinations. In *Proc. 4th International Conference on Multi-Agent Systems (ICMAS-00)*, 2000.

[Aus97] Lawrence M Ausubel. An efficient ascending-bid auction for multiple objects. Technical report, Department of Economics, University of Maryland, 1997.

[AC98] Lawrence M Ausubel and Peter Cramton. The optimality of being efficient. Technical report, University of Maryland, 1998.

[AM02] Lawrence M Ausubel and Paul R Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1:1–42, 2002.

[BN01] Moshe Babaioff and Noam Nisan. Concurrent auctions across the supply chain. In *Proc. 3rd ACM Conference on Electronic Commerce*, pages 1–10. ACM Press, 2001.

[BLP89] J S Banks, J O Ledyard, and D Porter. Allocating uncertain and unresponsive resources: An experimental approach. *The Rand Journal of Economics*, 20:1–25, 1989.

[BGN02] Y Bartal, R Gonen, and N Nisan. Incentive compatible multi unit combinatorial auctions. Technical report, The Hebrew University of Jerusalem, 2002.

[Bar89] J J Bartholdi. The computatonal difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.

[BW01] Damian R Beil and Lawrence M Wein. An inverse-optimization-based auction mechanism to support a multi-attribute RFQ process. Technical report, Operations Research Center, MIT, 2001.

[Ber87] Dimitri P Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, 1987.

[Ber88] Dimitri P Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14:105–123, 1988.

[BK02] Martin Bichler and Jayant Kalagnanam. Winner determination problems in multi-attribute auctions. Technical report, IBM Research report, 2002.

[BKL02] Martin Bichler, Jayant Kalagnanam, and Ho Soo Lee. RECO: Representation and Evaluation of Configurable Offers. Technical Report RC 22288, Jan, IBM Research report, 2002. to appear in Proceedings of ICS 2003.

[BdVSV01] Sushil Bikchandani, Sven de Vries, James Schummer, and Rakesh V Vohra. Linear programming and Vickrey auctions. Technical report, Anderson Graduate School of Management, U.C.L.A., 2001.

[BO00] Sushil Bikchandani and Joseph M Ostroy. Ascending price Vickrey auctions. Technical report, Anderson Graduate School of Management, U.C.L.A., 2000.

[BO02] Sushil Bikchandani and Joseph M Ostroy. The package assignment model. *Journal of Economic Theory*, 2002. forthcoming.

[BN02] Liad Blumrosen and Noam Nisan. Auctions with severely bounded communications. Technical report, The Hebrew University of Jerusalem, 2002. preliminary version.

[Bou02] Craig Boutilier. Solving concisely expressed combinatorial auction problems. In *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*, July 2002. to appear.

[BH01] Craig Boutilier and Holger Hoos. Bidding languages for combinatorial auctions. In *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.

[Bra97] F Branco. The design of multidimensional auctions. *RAND Journal of Economics*, 28:63–81, 1997.

[Bre99] Paul J Brewer. Decentralized computation procurement and computational robustness in a smart market. *Economic Theory*, 13:41–92, 1999.

[CS83] K Chatterjee and W Samuelson. Bargaining under incomplete information. *Operations Research*, 31:835–851, 1983.

[Che93] Y K Che. Design competition through multidimensional auctions. *RAND Journal of Economics*, 24:668–680, 1993.

[Cla71] E H Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[Cli97] D. Cliff. Minimial-intelligence agents for barganing behaviors in market-based environments. Technical Report Technical Report HPL-97-91, Hewlett Packard Labs, 1997.

[Cli01] D Cliff. Evolution of market mechanism through a continuous space of auction types. Technical Report Technical Report HPL-2001-326, Hewlett Packard Labs, 2001.

[CJ00] Olivier Compte and Philippe Jehiel. On the virtues of the ascending price auction: New insights in the private value setting. Technical report, UCL, 2000.

[CJ02] Olivier Compte and Philippe Jehiel. Auctions and information acquisition. Technical report, University College London, 2002.

[CS01] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions. In *Proc. 3rd ACM Conf. on Electronic Commerce (EC-01)*. ACM Press, New York, 2001.

[CS02a] V Conitzer and T Sandholm. Complexity of manipulating elections with few candidates. In *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*, July 2002. to appear.

[CS02b] V Conitzer and T Sandholm. Complexity of mechanism design. In *Proc. 18th Conf. on Uncertainty in Artificial Intelligence (UAI'02)*, 2002.

[CLSS01] Janos A Csirik, Michael L Littman, Satinder Singh, and Peter Stone. FAucS: An FCC Spectrum Auction Simulator for Autonomous Bidding Agents. In *Proc. 2nd Int. Workshop on Electronic Commerce*, 2001.

[DHKT01] R Das, J E Hanson, J O Kephart, and G Tesauro. Agent-human interactions in the continuous double auction. In *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*. 2001.

[HRN02] Abigail Hohner, John Rich, Ed Ng, Grant Reid, Andrew Davenport, Jayant Kalagnanam, Ho Soo Lee and Chae An. Combinatorial and quantity discount procurement auctions with mutual benefits at Mars, Inc. *Interfaces*, 2002. forthcoming.

[DK01] Andrew Davenport and Jayant Kalagnanam. Price negotiations for procurement of direct inputs. In Brenda Dietrich and Rakesh Vohra, editors, *Mathematics of the Internet: E-Auction and Markets*. IMA Volumes in Mathematics and its Applications, Springer-Verlag, 2001.

[DK02] Andrew Davenport, Abigail Hohner, and Jayant Kalagnanam. Iterative reverse combinatorial auctions with side constraints. IBM Research Report 2002.

[DH93] Douglas D. Davis and Charles A. Holt. *Experimental Economics*. Princeton University Press, 1993.

[DCK02] Milind Dawande, R. Chandrasekharan, and Jayant Kalagnanam. On a question in linear programming and its application in decentralized allocation. Technical report, IBM research report, jul 2002.

[dVV02] Sven de Vries and Rakesh V Vohra. Combinatorial auctions: A survey. *Informs Journal on Computing*, 2002. forthcoming.

[DGS86] Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[DKLP98] C DeMartini, A M Kwasnica, J O Ledyard, and D Porter. A new and improved design for multi-object iterative auctions. Technical Report SSWP 1054, California Institute of Technology, 1998. Revised March 1999.

[EK02] Wedad Elmaghraby and Pinar Keskinocak. Combinatorial Auctions in Procurement. Technical report, National University of Singapore, Januaury 2002.

[FKSS01] Joan Feigenbaum, Arvind Krishnamurthy, Rahul Sami, and Scott Shenker. Approximation and collusion in multicast cost sharing.

In *Proc. of the 3rd Conference on Electronic Commerce*, pages 253–255, 2001.

[FPSS02] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the 2002 ACM Symposium on Principles of Distributed Computing*, 2002. to appear.

[FPS01] Joan Feigenbaum, Christos H Papadimitriou, and Scott Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63:21–41, 2001.

[Fri91] Daniel Friedman. Evolutionary games in economics. *Econometrica*, 59(3):637–666, 1991.

[FLBS99] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553, 1999.

[GW00] Jeremie Gallien and Lawrence Wein. Design and analysis of a smart market for industrial procurement. Technical report, Operations Research Center, MIT, 2000.

[GJ79] Michael R Garey and David S Johnson. *Computers and Intractability*. W.H.Freeman and Company, 1979.

[Gib73] Alan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–602, 1973.

[GZ89] I Gilboa and E Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, pages 213–221, 1989.

[GJJ77] Jerry Green and Jean-JacquesLaffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.

[Gro73] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[GS00] Faruk Gul and Ennio Stacchetti. The English auction with differentiated commodities. *Journal of Economic Theory*, pages 66–95, 2000.

[HB00] Holger H Hoos and Craig Boutilier. Solving combinatorial auctions with stochastic local search. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, July 2000.

[HS02] B Hudson and T Sandholm. Effectiveness of preference elicitation in combinatorial auctions. In *Proc. Agent-Mediated Electronic Commerce (AMEC'IV) workshop*, 2002. at AAMAS-02.

[Jac00] Matthew Jackson. *Mechanism Theory*. 2000. Forthcoming in Encyclopedia of Life Support Stystems.

[KDL01] Jayant R Kalagnanam, Andrew J Davenport, and H S Lee. Computational aspects of clearing continous call double auctions with assignment constraints and indivisible demand. *Electronic Commerce Journal*, 1(3):221–238, 2001.

[KLS01] Michael Kearns, Michael L Littman, and Satinder Singh. Graphical models for game theory. In *Proc. of UAI*, 2001.

[KC82] Alexander S Kelso and Vincent P Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50:1483–1504, 1982.

[KG02] J Kephart and A Greenwald. Shopbot economics. *Autonomous Agents and Multiagent Systems*, 2002. forthcoming.

[KP97] D Koller and A J Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94:167–215, 1997.

[KM01] Daphne Koller and Brian Milch. Multi-agent influence diagrams for representing and solving games. In *Seventeenth International Joint Conference on Artificial Intelligence*, pages 1027–1034, 2001.

[Koz92] J R Koza. *Genetic Programming: On the programming of computers by means of natural selection*. MIT Press, 1992.

[KP98] Vijay Krishna and Motty Perry. Efficient mechanism design. Technical report, Pennsylvania State University, 1998. Available at: http://econ.la.psu.edu/˜vkrishna/vcg18.ps.

[LOP$^+$00] John O Ledyard, Mark Olson, David Porter, Joseph A Swanson, and David P Torma. The first use of a combined value auction for transportation services. Technical Report Social Science Working Paper 1093, California Institute of Technology, 2000.

[LOS99] Daniel Lehmann, Liadan O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proc. 1st ACM Conf. on Electronic Commerce (EC-99)*, pages 96–102, 1999.

[LWJ00] A R Lomuscio, M Wooldridge, and N R Jennings. A classification scheme for negotiation in electronic commerce. In F Dignum and C Sierra, editors, *Agent mediated electronic commerce: a European perspective*, pages 19–33. Springer-Verlag, 2000.

[MT80] Silvano Martello and Paulo Toth. *Knapsack Problems*. Wiley-Intersciences Series in Discrete Mathematics and Optimization, 1980.

[McA92] R Preston McAfee. A dominant strategy double auction. *J. of Economic Theory*, 56:434–450, 1992.

[McM94] John McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 8:145–62, 1994.

[Mil02] Paul Milgrom. *Putting Auction Theory to Work*. MIT Press, 2002.

[MT01] Dov Monderer and Moshet Tennenholtz. Bundling equilibrium. Technical report, Technion, 2001.

[Mye81] Robert B Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981.

[MS83] Robert B Myerson and Mark A Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 28:265–281, 1983.

[Nis00] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, pages 1–12, 2000.

[NR00] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, pages 242–252, 2000.

[NR01] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[NS02] Noam Nisan and I Segal. The communication complexity of efficient allocation problems. Technical report, Hebrew University and Stanford University, 2002.

[OR94] Martin J Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[Pap01] C H Papadimitriou. Algorithms, games and the Internet. In *Proc. 33rd Annual ACM Symp. on the Theory of Computing*, pages 749–753, 2001.

[Par99a] David C Parkes. *i*Bundle: An efficient ascending price bundle auction. In *Proc. 1st ACM Conf. on Electronic Commerce (EC-99)*, pages 148–157, 1999.

[Par99b] David C Parkes. Optimal auction design for agents with hard valuation problems. In *Proc. IJCAI-99 Workshop on Agent Mediated Electronic Commerce*, pages 206–219, July 1999. Stockholm.

[Par01] David C Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, May 2001. http://www.cis.upenn.edu/˜dparkes/diss.html.

[Par02] David C Parkes. Price-based information certificates for minimal-revelation combinatorial auctions. In *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems Workshop on Agent-Mediated Electronic Commerce*, Bologna, Italy, 2002.

[PK02] David C Parkes and Jayant Kalagnanam. Iterative multiattribute Vickrey auctions. Technical report, Harvard University, 2002. submitted for publication.

[PKE01b] David C Parkes, Jayant R Kalagnanam, and Marta Eso. Vickrey-based surplus distribution in combinatorial exchanges. In *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.

[PU00a] David C Parkes and Lyle H Ungar. Iterative combinatorial auctions: Theory and practice. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 74–81, July 2000.

[PU00b] David C Parkes and Lyle H Ungar. Preventing strategic manipulation in iterative auctions: Proxy agents and price-adjustment. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 82–89, July 2000.

[PU02] David C Parkes and Lyle H Ungar. An ascending-price generalized Vickrey auction. Technical report, Harvard University, 2002.

[PMPS02] S Phelps, P McBurney, S Parsons, and E Sklar. Co-evolutionary auction mechanism design: A preliminary report. In *Proc. AMEC'IV Workshop, at AAMAS'02*, 2002.

[PMM87] R Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, June 1987.

[Pri97] T C Price. Using co-evolutionary programming to simulate strategic behavior in markets. *Journal of Evolutionary Economics*, 7:219–254, 1997.

[RSB82] S J Rassenti, V L Smith, and R L Bulfin. A combinatorial mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982.

[RWG01] Daniel M Reeves, Michael P Wellman, and Benjamin N Grosof. Automated negotiation from declarative descriptions. In *Proc. 5th International Conference on Autonomous Agents (AGENTS-01)*. 2001.

[RSCL02] Paul S A Reitsma, Peter Stone, Janos A Csirik, and Michael L Littman. Self-enforcing strategic demand reduction. In *Proc. Agent Mediated Electronic Commerce IV (AMEC'IV)*, 2002.

[Ron01] Amir Ronen. Mechanism design with incomplete languages. In *Proc. 3rd ACM Conf. on Electronic Commerce (EC'01)*, 2001.

[Rot02] A E Roth. The economist as engineer. *Econometrica*, 70:1341–1378, 2002.

[RM82] A E Roth and J K Murnighan. The role of information in bargaining: An experimental study. *Econometrica*, 50:1123–1142, 1982.

[RPH98] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[RMP92] J Rust, J Miller, and R Palmer. Behavior of trading automata in a computerized double auction market. In D Friedman and J Rust, editors, *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Welsey, 1992.

[RMP94] J Rust, J Miller, and R Palmer. Characterizing effective trading strategies: Insights from the computerized double auction tournament. *J. of Ecomomic Dynamics and Control*, 18:61–96, 1994.

[San00] T Sandholm. Agents in electronic commerce: Component technologies for automated negotiation and coalition formation. *Autonomous Agents and Multi-Agent Systems*, 3(1):73–96, 2000.

[San96] Tuomas W Sandholm. Limitations of the Vickrey auction in computational multiagent systems. In *Second International Conference on Multiagent Systems (ICMAS-96)*, pages 299–306, 1996.

[San00] Tuomas W Sandholm. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4:107–129, 2000.

[SS01a] Tuomas W Sandholm, Subhash Suri. Side Constraints and Non-Price Atributes in Markets In *Int. Joint Conf. on Artificial Intelligence, Workshop on Distributed Constraint Reasoning*, 2001.

[SS01b] Tuomas W Sandholm, Subhash Suri. Market Clearability In *Proc. of Int. Joint Conf. on Artificial Intelligence*, 2001.

[SSGL01a] Tuomas W Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proc. 17th Int. Joint Conf. on Artificial Intelligence*, 2001.

[SSGL01b] Tuomas W Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. Winner determination in combinatorial auction generalizations. In *Proc. International Conference on Autonomous Agents, Workshop on Agent-based Approaches to B2B*, pages 35–41, 2001.

[Sat75] Mark A Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[SW89] Mark A Satterthwaite and Steven R Williams. Bilateral trade with the sealed bid k-double auction: Existence and efficiency. *Journal of Economic Theory*, 48:107–133, 1989.

[Sch01] James Schummer. Almost dominant strategy implementation. Technical report, MEDS Department, Kellogg Graduate School of Management, 2001.

[Smi62] V L Smith. An experimental study of competitive market behavior. *J. of Political Economy*, 70:111–137, 1962.

[Smi82] V L Smith. Microeconomic systems as an experimental science. *American Economic Review*, 72:923–955, 1982.

[TWD02] Valentina Tamma, Michael Wooldridge, and Ian Dickinson. An ontology based approach to automated negotiation. In *Proc. AMEC'IV Workshop, at AAMAS'02*, 2002.

[TKDM00] Moshe Tennenholtz, Noa Kfir-Dahav, and Dov Monderer. Mechanism design for resource bounded agents. In *Proc. 4th International Conference on Multi-Agent Systems (ICMAS-00)*, 2000.

[TD01] Gerald Tesauro and Rajarshi Das. High-performance bidding agents for the continuous double auction. In *Proc. 3rd ACM Conf. on Electronic Commerce (EC-01)*. ACM Press, New York, 2001.

[Var95] Hal R Varian. Economic mechanism design for computerized agents. In *Proc. USENIX Workshop on Electronic Commerce*, 1995. Minor update, 2000.

[Var02] Hal R Varian. When economics shifts from science to engineering. *New York Times*, August 29 2002. Economic Scene.

[Vic61] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[WWO$^+$01] M P Wellman, P R Wurman, K O'Malley, R Bangera, S Lin, D Reeves, and W E Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, pages 43–51, 2001.

[WGSW02] Michael P Wellman, Amy Greenwald, Peter Stone, and Peter R Wurman. The 2001 Trading Agent Competition. In *Proc. 14th Conf. on Innovative Applications of Art. Intell.*, pages 935–941, 2002.

[Wil99] Steven R Williams. A characterization of efficient, Bayesian incentive-compatible mechanisms. *Economic Theory*, 14:155–180, 1999.

[Wil85] R Wilson. Incentive efficiency of double auctions. *Econometrica*, 53:1101–1115, 1985.

[WWW98] Peter R Wurman, William E Walsh, and Michael P Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.

[WWW01] Peter R Wurman, Michael P Wellman, and William E Walsh. A parameterization of the auction design space. *Games and Economic Behavior*, 35:304–338, 2001.

[WWW02] Peter R. Wurman, Michael P. Wellman, and William E. Walsh. Specifying rules for electronic auctions. *AI Magazine*, 2002.

[Yoo01] Kiho Yoon. The modified vickrey double auction. *Journal of Economic Theory*, 2001.

[ZW02] Weili Zhu and Peter R. Wurman. Structural leverage and fictitious play in sequential auctions. In *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*, July 2002.