Auctions, Bidding and Exchange Design

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 multi-attribute 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 Section 2 we first introduce the economic literature on *mechanism design* and identify the economic properties that are desirable in the design of auction mechanisms. Continuing, in Section 3 we introduce the associated computational complexities that arise in the implementation of optimal mechanisms. Section 4 provides an extended discussion of the subtle interaction between computation and incentives. Then, in Section 5 we pick off a few specific mechanisms that are interesting both from a practical point of view and also because they illustrate some of the emerging research directions. Finally, Section 6 concludes with a brief discussion of the role of mechanism design in the design of electronic markets and considers a new computational approach, termed *experimental* mechanism design.

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. On the other hand, 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.

We begin with a description of mechanism design theory and competitive equilibrium theory. Then, we outline a primal-dual approach to the design of indirect mechanisms (such as ascending price-based auctions), that unifies the mechanism design and competitive equilibrium approaches. Essentially, one can view an ascending-price auction as a method to implement a primal-dual algorithm for a resource allocation problem. Terminating with competitive equilibrium prices that also implement the outcome of a mechanism brings the price-taking behavior that is assumed in classic competitive equilibrium behavior into a game-theoretic 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.

[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].

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 valuation 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 particular 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 [AC99], 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 [DHK02]. 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, $g : \Theta \to \mathcal{K}$, and a payment rule, $p : \Theta \to \mathbb{R}^N$. 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:

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

[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.

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.

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.

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

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

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

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:

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

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:

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

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:

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

where $g_{\text{eff}}(\hat{\theta}_{-i})$ 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

[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.

result states that

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

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]

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*. The Myerson-Satterthwaite result states that 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 (typically the seller). 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

[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, 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.

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, $g : \Theta \to \Delta(\mathcal{K})$, defines a probability distribution over choices given reported types, and the payment rule, $p : \Theta \to \mathbb{R}^N$, 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.

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

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

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

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

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 de Vries & Vohra 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):

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

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

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

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:

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

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:

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

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

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

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

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.

To illustrate competitive equilibrium (CE) prices we will impose some structure on choice set $\mathcal{K}$ and define the *combinatorial allocation problem* (CAP). Let $\mathcal{G}$ define a set of items, and $S \subseteq \mathcal{G}$ 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. Agent i has value $v_i(S)$ for bundle S.[5] Assume for the purpose of exposition that we have knowledge of agent valuations. The CAP can be

[5] This implies a valuation $v_i(k, \theta_i) = v_i(S')$ on allocation k, where $S' = \cup_{(S,i) \in k} S$ is the union of all bundles allocated to agent i in allocation k.

formulated as the following integer program:

$$\max_{x_i(S)} \sum_{S \subseteq \mathcal{G}} \sum_{i \in \mathcal{I}} x_i(S) v_i(S) \qquad \text{[CAP]}$$

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

$$\sum_{S \ni j} \sum_{i \in \mathcal{I}} x_i(S) \leq 1, \quad \forall j \in \mathcal{G}$$

$$x_i(S) \in \{0, 1\} \qquad (1.6)$$

where $S \ni j$ indicates that bundle S contains item j. Later, we find it useful to use CAP($\mathcal{I}$) to denote the CAP with all agents $\mathcal{I}$ and CAP($\mathcal{I} \setminus j$) to denote the CAP formulated without agent $j \in \mathcal{I}$.

To apply linear-programming duality theory we must relax this IP formulation, and construct an integral LP formulation. Consider [LP$_1$] in which eq. (1.6) is relaxed to $x_i(S) \geq 0$. Then, the dual is simply written as:

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

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

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

The dual 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. The dual problem computes prices on items to minimize the sum of the payoffs across all agents. These are precisely a set of CE prices when the primal solution is integral.

A technical condition, *gross substitutes* [KC82] (or simply substitutes) on agent valuations is sufficient for integrality. Unit demand preferences, in which each agent wants at most one item (but can have different values for different items), is a special case. With this substitutes condition, LP$_1$ is integral and the complementary-slackness (CS) conditions on a feasible primal, x, and feasible dual, p, solution define conditions for competitive equilibrium:

$$\pi_i > 0 \Rightarrow \sum_S x_i(S) = 1, \quad \forall i \qquad (1.7)$$

$$p(j) > 0 \Rightarrow \sum_{S \ni j} \sum_i x_i(S) = 1, \quad \forall j \qquad (1.8)$$

$$x_i(S) > 0 \Rightarrow \pi_i + \sum_{j \in S} p(j) = v_i(S), \quad \forall i, \forall S \qquad (1.9)$$

These have a natural economic interpretation. Conditions (1.7) and (1.9) state that the allocation must maximize the payoff for every agent at the prices. Condition (1.8) states that the seller must sell every item with a positive price, and maximize the payoff to the seller at the prices. The prices are said to *support* the efficient allocation. 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. In practice we will need an auction to provide incentives for agents to reveal the information about their valuations, and to converge towards a set of CE prices.

The linear program formulation, LP_1, is *not* integral in general instances of CAP and the item prices will not support the efficient allocation. Instead one needs to consider prices on bundles of items. Bikhchandani & Ostroy [BO02b] provide a hierarchy of strengthened LP formulations to capture these generalizations, in which the variables in the dual problems correspond with *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 $i \neq j$. We will return to these extended formulations and enriched price space in Section 5.1 in reference to the design of an ascending-price combinatorial auction.

2.4 Indirect Revelation Mechanisms

In this section, we tie together the mechanism design approach and the competitive equilibrium approach. The basic idea is to construct efficient ascending-price auctions that terminate with the outcome of the VCG mechanism. With this, price-taking behavior is a game-theoretic equilibrium of the auction despite the effect that an agent's bids might have on future price dynamics. The auctions provide a dynamic method to compute a set of competitive equilibrium prices, from which allocative efficiency follows.

To understand the motivation for the design of iterative, price-based mechanisms, we need to begin to consider the computational considerations in the implementation of useful auction mechanisms. Although the revelation principle focuses attention on incentive-compatibility, it hides all implementation and computational concerns. In particular, a direct revelation mechanism (such as the VCG) requires every agent to provide complete and exact information about its valuation over all possible outcomes. This is often unreasonable, for example in the setting of a combinatorial auction to allocate the rights to operate the 700+ bus routes in London.[6] In comparison, an agent in an iterative price-based auction can reveal its preference information as necessary.

These kinds of mechanisms, in which agents are *not* required to submit (and compute) complete and exact information about their private valuations, are referred to as *indirect* mechanisms. Indirect mechanisms, such as those based on prices, also go some way to distributing the calculation of the outcome of a mechanism across agents rather than requiring the mechanism infrastructure (such as the auctioneer) to compute the winners and the payments.

Examples of indirect mechanisms include *ascending-price auctions*, in which agents submit bids in response to prices and the auctioneer maintains a provisional allocation and adjusts prices. For example, the English auction is an ascending-price auction for a single item in which the price increases until there is only one bidder left in the auction [PMM87]. The English auction implements the outcome of the Vickrey auction (and is allocatively-efficient). However, only two agents must bid to make progress towards the outcome, and agents can follow equilibrium strategies with lower- and upper-bounds on their values [Par99b, CJ00]. In comparison, the equilibrium in the Vickrey auction requires every agent to compute, and reveal, exact information about its value.

We describe a general methodology to design iterative mechanisms that leverages a fundamental connection between linear programming, competitive equilibrium, and VCG payments. The approach has been used in recent years to design and analyze efficient ascending auctions for the assignment problem [DGS86], combinatorial auctions [PU00a, PU02], multiattribute auctions [PK03], and multi-unit auctions [BdVSV01]. The interested reader is also referred to Bikhchandani & Ostroy [BO02a] and Parkes [Par01] for an extended discussion.

In outline, the two steps in a primal-dual approach to the design of efficient ascending auctions are:

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.

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 any additional conditions neces-

[6] Indeed, a *limited* combinatorial auction, in which operators can submit a restricted number of bids, has operated for the competitive tendering of London bus routes since 1997 [CP03].

sary to compute the VCG payments. The algorithm should not assume complete access to agent valuations, but rather access to myopic best-response bids from agents.

Note carefully that termination in VCG payments is sufficient to bring myopic best-response into a game-theoretic equilibrium. As such, the first assumption is used to leverage the primal-dual design methodology, but is not limiting from an incentive perspective. Technically, MBR is an *ex post* Nash equilibrium, in which there is no better strategy for an agent whatever the preferences of other agents, so long as the other agents also follow myopic best-response [GS00, PU02]. *Ex post* Nash 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.

In the special case of *agents-are-substitutes*, then the *minimal* CE prices support the VCG payments to each agent. The minimal CE prices are a set of prices that *minimize* the revenue to the seller (or equivalently, maximize the total payoff to the buyers) across all CE prices (and need not be unique in general). Let *coalitional value*, $w(L) \geq 0$, denote the value of the efficient allocation for agents $L \subseteq \mathcal{I}$, i.e. the solution to CAP($\mathcal{L}$). Agents-are-substitutes places the following constraints on the coalitional values:

$$w(\mathcal{I}) - w(\mathcal{I} \setminus K) \geq \sum_{i \in K} \left[w(\mathcal{I}) - w(\mathcal{I} \setminus i) \right], \quad \forall K \subseteq \mathcal{I}$$

This is known to be the widest class of preferences for which the VCG outcome can be supported in a competitive equilibrium, even with non-linear and non-anonymous prices [BO02b]. It holds, for example when we have *gross substitutes* preferences [AM02], and in the unit-demand problem. When agents-are-substitutes fails, the minimal CE prices are not unique, and moreover they do not support the VCG outcome.

However, the primal-dual methodology does *not* require that the VCG payments are supported in a single set of CE prices. Rather, it requires that the price space is rich enough to support *Universal* competitive equilibrium (UCE) prices. From these prices, we can compute the VCG payments [PU02]. UCE prices are defined as CE prices that are simultaneously CE prices for CAP($\mathcal{I}$), and also for CAP($\mathcal{I} \setminus j$) without each agent $j \in \mathcal{I}$. UCE prices always exist, for example setting $p_i(S) = v_i(S)$ for all agents provides UCE prices. In the case of agents-are-substitutes, then minimal CE prices are UCE prices. Given UCE prices, p_{uce}, we can compute the VCG payments to each agent as:

$$p_{\text{vcg},i} = p_{\text{uce}}(S_i^*) - \left[\Pi(p_{\text{uce}}, \mathcal{I}) - \Pi(p_{\text{uce}}, \mathcal{I} \setminus i) \right] \tag{1.10}$$

where $\Pi(p, L)$ is the maximal revenue that the auctioneer can achieve, given prices p, across all feasible allocations to agents in set $L \subseteq \mathcal{I}$.

We return to UCE prices in Section 5.1, in the context of an ascending-price combinatorial auction in which agents are interested in bundles of items. The primal-dual analysis is performed with respect to the hierarchy of extended LP formulations described in Section 2.3.

2.4.1 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:

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

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.

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

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:

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

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 $\hat{i}$. Construct a feasible primal solution with $y = 0$, $x_{\hat{i}} = 1$ and $x_i = 0$ for all $i \neq \hat{i}$. 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 *minimal* CE prices, or the dual solution that maximizes the payoff to the winning agent across all 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 $\epsilon \rightarrow 0$. In this simple auction this is sufficient to make MBR a *dominant* strategy for an agent together with a rule that prevents jump bids.

3. Implementation Considerations

In this section, we continue to discuss some of the computational considerations that must be addressed in taking a mathematical specification of a mechanism and building a working system. There often remains a large gap between the mathematical specification and a reasonable computational implementation. We have already introduced the idea of *indirect mechanisms* that can reduce the amount of information that agents must provide about their preferences. Here, in laying out some of the additional details that must be considered in closing this gap, we consider the choice of a language to represent agent preferences (Section 3.1) and the complexity of the winner-determination problem (Section 3.2). The winner-determination complexity can also be impacted by side constraints that represent business rules (Section 3.2.2).

3.1 Bidding Languages

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.

$\mathbb{L}_B$ [San02, Nis00] languages use bundles with associated prices as atomic propositions and combines them using logical connectives. One bid language is **additive-or**, $\mathbb{L}_B^{OR}$, in which one or more bids can be accepted, and the total bid price is the sum of the individual bid prices. This language is compact for particular valuations (e.g. linear-additive across items), but not expressive for general valuations. The canonical language is the **exclusive-or**, $\mathbb{L}_B^{XOR}$, language in which bids state that *at most one* bid can be accepted. The total bid price is the value of the maximal bid price across the component bundles when multiple bundles are accepted. One can also consider nested languages, such as OR-of-XORs and XOR-of-ORs, and a generalization, $\mathbb{L}_B^{OR*}$, in which dummy goods are used within atomic bids to provide expressiveness with more compactness than either OR-of-XOR or XOR-of-OR. 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. These generalized logical bids inherit the advantages of both the atomic bid and the atomic bundle 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

In the context of auctions, the problem of computing an allocation is often referred to as the *winner-determination* problem. In simple designs (such as the English, Vickrey 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 four cases: (i) divisible bids, (ii) indivisible bids with XOR bid structures, (iii) price schedules, which can be viewed as a compact representation for a generalized XOR with indivisible bids. Suppose in all of these cases that a buyer wants to buy Q identical units of the same item.

Divisible Bids. In the simple case, each bidder submits a price-quantity pair (p_i, q_i), to indicate that it will sell up to q_i units for a unit price of p_i/q_i. 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.

Indivisible Bids. Now, suppose that bidders specify *all-or-nothing* constraints on the bids and state that the bids are indivisible. In addition, suppose that the bidders also submit multiple bids with an XOR bidding language. 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, SS01]. 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, $\{(p_{i1}, [\underline{q}_{i1}, \overline{q}_{i1}]), \ldots (p_{iM_i}, [\underline{q}_{iM_i}, \overline{q}_{iM_i}])\}$. 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.

Recently, Kothari *et al.* [KPS03] have proposed a fully polynomial-time approximation scheme (FPTAS) for a variation on this price-schedule problem in which the cost functions are piecewise and *marginal-decreasing* and each supplier has a *capacity constraint*. The approach is to construct a 2-approximation to a generalized knapsack problem, which can then be used to scale a dynamic-programming algorithm and compute an $(1 + \epsilon)$ approximation in worst-case time $T = O((nc)^3/\epsilon)$, for n bidders and with a maximum of c pieces in each bid.[7]

3.2.2 Multi-Item Auctions.

In this subsection we introduce multi-item auctions for multiple heterogenous items. This is the well known combinatorial auction problem, in which we allow bidders to have arbitrary valuations over bundles of items.

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 and submit bids for multiple bundles via an

[7] Interestingly, it is also possible to carefully combine two dynamic programming tableaus in order to approximate the VCG payments for an asymptotic cost of $O(T \log n)$, instead of the typical additional worst-case cost of a factor of $O(n)$.

XOR bidding language. 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.

Forward auction. For the forward auction case of a *single seller with multiple buyers*, the winner-determination problem can 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. Many 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, real world problems will often not satisfy the fairly severe restrictions that provide a totally unimodular constraint matrix. Moreover, if the bidding language is not expressive then this can interact with the incentive properties of an auction because a bidder is not able to express her true valuation, even if that would be her equilibrium strategy. We wait until Section 4 for an extensive discussion of the interaction between computational constraints and incentives.

Reverse auction. Combinatorial auctions are also proposed for procurement problems in markets with one buyer and multiple sellers. The reverse combinatorial auction is formulated as a set covering problem rather than a set packing problem. 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. The winner determination problem with no side constraints can be written as:

$$\min_{x_i(S)} \sum_{S \in B_i} \sum_i x_i(S) p_i(S) \tag{1.11}$$

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

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

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

for bids in set $B_i \subseteq 2^{\mathcal{G}}$ with prices $p_i(S)$ on $S \in B_i$. This is posed as a cost minimization problem with a demand covering constraint. In this formulation the problem is procure a single unit of each good, but this can be generalized by increasing the RHS of the first set of constraints.

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 satisfied while picking a set of winning suppliers. Recent work [DK01, SSGL01b, BK02] in this area provides a comprehensive overview of the constraint types that are possible. We discuss some of the main constraint classes here, and provide some example MIP formulations in the context of (1.11). In general, the specific form of these side constraints depends on the market structure.

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, L_s, and maximum, U_s, number of winning suppliers in the solution to the winner determination problem.

$$y_i \leq \sum_{S \in B_i} x_i(S) \leq K y_i, \quad \forall i \in N$$

$$L_s \leq \sum_i y_i \leq U_s$$

Budget Limits on Trades A common constraint that is often placed is a upper limit on the total volume of the transaction with a 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.

Reservation Prices A reservation price allows the buyer to place an additional constraint on the most she will pay for some items. This can arise, for example, due to a fall-back option such as an external commodity market. If the reservation prices are specified over bundles then

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

where $r(S)$ is the reservation price. Alternately, reservation prices can be specified for each item, with $r(S) = \sum_{j \in S} r(j)$, and $r(j)$ to define the price on item j.

Representation Constraints These specify additional 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.

The interesting aspect of these side constraints is how they impact the computational complexity of the winner determination problem. For example, introducing either of the following constraint classes will transform even a tractable problem (e.g. with a totally unimodular structure) into a hard problem:

- Budget constraints with integrality requirements for the choice of bids lead 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.

Volume Discounts. So far we have treated the price on a bundle as an all-or-nothing bid. An alternative is to consider *supply curves*, in which the price function is explicitly specified in terms of volume discounts for each item over which a total price is constructed for a given bundle. It is useful to assume that the supply curves are additive separable; that is,

$$p_i(A^s) = p_i(a_1^s, .., a_M^s) = \sum_{j \in \mathcal{G}} p_{ij}(a_i^s)$$

where p_{ij} are individual price curves for the commodity j from supplier i, and a_i^s is the quantity for commodity j. We can also assume that each individual curve is a piece-wise linear function.

The winner determination for supply curve auctions can also be written as a set covering problem as shown Eq. (1.11) using a Dantzig-Wolfe type decomposition [EGKL01]. To use a set covering model we introduce the concept of supply patterns. A supply pattern A^s is a vector of length M specifying the amount supplied for each of the commodities $A^s = (a_1^s, a_2^s, ..., a_k^s)$. The cost of a supply pattern for a particular supplier is computed as $p_i(A^s)$. A supply pattern is feasible for a supplier if she is able to sell the given amount from each of the commodities, and meet additional side constraints. The set of feasible supply patterns for supplier i is denoted by $\mathcal{D}_i$. 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, $s \in \bigcup_i \mathcal{D}_i$.

Let Q_j denote the quantity of commodity j demanded in the procurement problem. The basic constraints of this optimization problem will ensure that the demand is met and that at most one pattern is chosen for each supplier:

$$\sum_i \sum_{s \in \mathcal{D}_i} a_j^s y^s \geq Q_j, \quad \forall j$$

$$\sum_{s \in \mathcal{D}_i} y^s \leq 1, \quad \forall i$$

A lower, L_s, and upper limit, U_s, for the total number of accepted suppliers, can again be imposed by the following constraint:

$$L_s \leq \sum_i \sum_{s \in \mathcal{D}_i} y^s \leq U_s$$

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_{ij} and u_{ij} are such limits for a particular supplier and commodity and that L_i and U_i are limits for a supplier across all commodities. Then any feasible pattern

$s \in \mathcal{D}_i$ for supplier j must satisfy the following constraints:

$$l_{ij} \leq a_j^s \leq u_{ij}, \quad \forall j$$
$$L_i \leq \sum_j a_j^s \leq U_i$$

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

$$z = \min \sum_i \sum_{s \in \mathcal{D}_i} p_i(A^s) y^s$$

3.2.3 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,[8] and (ii) the clearinghouse or *call* auction, which clears periodically. In this Chapter we focus on call markets. Call markets have received more attention in the literature, and are perhaps more appropriate when bids and asks are combinatorial because collecting a number of bids before clearing the market can improve the ability to construct useful matches between complex bids and asks.

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.

[8] 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.

The appropriate level of aggregation depends on the *physical attributes* of the good. For example, 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). Note that aggregation *does not* imply that the exchange must take physical possession of goods, trades can still be executed directly between agents.

Similarly, goods can have multiple attributes, and must often be considered as 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.

Our discussion considers the case of bids and asks for multiple units of the same item, that we term *homogeneous bids*, and the *combinatorial exchanges* in which bids and asks can bundle together multiple units of multiple different items.

Homogeneous Bids. First, we consider the case of bids and asks for multiple units of the same item, but without allowing bundle bids. Moreover, we assume that multiple bids and asks submitted by the same bidder are connected with *additive-or* logic. For the moment we also assume that bids and asks are divisible, so that a fraction of a bid can be matched with an ask (or multiple fractions with multiple asks). We provide a general formulation of the winner determination problem in this setting. The formulation captures different market structures, in terms of aggregation and differentiation.

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, each ask, $a_j \in A$, is associated with a single type of good, and provides a unit ask price, p_j, and a quantity offered, q_j. 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 $B_j \subseteq B$ to which it can be feasibly matched. These assignment restrictions model the feasibility requirements imposed by the heterogeneity of goods. We do not specify the constraints in any more detail because this depends on the structure of the market. Then, the winner-determination problem, to clear the exchange to maximize surplus, is written as:

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

$$\text{s.t.} \quad \sum_{i \in B_j} q_i x_{ij} \leq q_j, \quad \forall j \in A \tag{1.12}$$

$$\sum_{j \in A_i} x_{ij} \leq 1, \quad \forall i \in B \tag{1.13}$$

$$0 \leq x_{ij} \leq 1 \quad , \forall i, j \tag{1.14}$$

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.

We can include assignment restrictions (for example to capture the case of heterogeneous goods) and still use a LP to solve the matching problem as long as bids are divisible and the logic connecting bids is 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.

However, if the bids are indivisible we just 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 (1.14) with $x_{ij} \in \{0, 1\}$. Still, if we now 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 (1.12) can be replaced with $\sum_{i \in B_j} x_{ij} \leq 1$ But, in general, for example with aggregation on the sell side, the constraint (1.13) 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.

Combinatorial Exchanges. In a combinatorial exchange we allow bids and asks on bundles of heterogeneous items, and allow a bidder to connect multiple bids and asks with an exclusive-or bidding language. We choose to formulate the problem for agents that either act exclusively as sellers or exclusively as buyers, but this is not necessary in general. The formulation of the market clearing problem generalizes the winner-determination problem for the one-sided combinatorial auction to allow multiple buyers and sellers.

Let $\mathcal{B}$ denote the set of buyers and $\mathcal{S}$ denote the set of sellers. We allow sellers to submit asks for multiple bundles $S \subseteq \mathcal{G}$, with an ask price, $m_i(S)$, on each bundle. Similarly, we allow buyers to submit bids for multiple bundles, with a bid price, $p_i(S)$ on each bundle. Let $i \in \mathcal{B} \cup \mathcal{S}$ index both buyers and sellers, and let $B_i \subseteq 2^{\mathcal{G}}$ denote the set of bundles that receive a bid (or ask) from agent i. Finally, variable $x_i(S) = 1$ indicates the bid on bundle S from buyer i is accepted, and $y_i(S) = 1$ indicates that the ask on bundle S from seller i is accepted. Given this, we can formulate the market clearing problem as:

$$\max_{x_i(S),y_i(S)} \sum_{S \in B_i} \sum_{i \in \mathcal{B} \cup \mathcal{S}} (x_i(S)p_i(S) - y_i(S)m_i(S))$$

$$\text{s.t.} \quad \sum_{S \in B_i} x_i(S) \leq 1, \qquad \forall i \in \mathcal{B}$$

$$\sum_{S \in B_i} y_i(S) \leq 1, \qquad \forall i \in \mathcal{S}$$

$$\sum_{S \in B_i, S \ni j} \sum_{i} (y_i(S) - x_i(S)) \geq 0, \qquad \forall j$$

$$x_i(S) \in 0,1, y_i(S) \in \{0,1\}$$

Although the general problem is NP-hard, the special case in which there is no aggregation is still equivalent to the assignment problem, and solvable via an LP formulation. 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.

3.2.4 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 K_j to denote the domain of attribute j (assumed discrete), and $K = K_1 \times \ldots \times K_m$ denote the joint domain, with $m = |\mathcal{J}|$. Consider a reverse auction setting, and write $v^B(x) \geq 0$ and $c_i(x) \geq 0$ to denote the buyer's value and the cost of seller i for attribute bundle $x \in K$. Of course, enumerating these valuations and cost functions over the cross-product of attribute levels can be costly for participants in a market.

Thus, it is useful to consider the *preferential-independence* (PI), in which an agent can state its value for different levels of a particular attribute irrespective of the levels of another attribute. In this special case, the valuation, $v^B(x) = \sum_{j \in \mathcal{J}} v^B_j(x_j)$, where $v^B_j(x_j) \geq 0$ is the buyer's valuation for level $x_j \in K_j$ of attribute j. Similarly, the cost, $c_i(x) = \sum_{j \in \mathcal{J}} c_{ij}(x_j)$, where $c_{ij}(q_j)$ is the supplier's cost for level $x_j \in K_j$ of attribute j.

We will first consider the *single-sourcing* problem, in which a single winning bid is selected to satisfy the demand. Then, we will consider the *multiple-sourcing* problem, in which there are multiple units to procure and a buyer is interested in purchasing from multiple sellers.

Single Sourcing. In a single-sourcing setting only a single winning bid is picked to satisfy the demand. Consider the case of preferential-independence, and let b_{ijk} denote the ask price from supplier i on level $k \in K_j$ of attribute j. Similarly, let v_{jk} denote the reported value of the buyer on level k of attribute j. 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 variable y_i is used to indicate that supplier i is selected in the winning allocation. A straightforward method to solve this problem computes the best attribute values for each supplier, and then chooses the best supplier.

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 are infeasible. Similarly, promotions to encourage certain attribute levels can be specified as rules. Propositional logic can be 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 configurable 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. 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 surplus (value - cost), and then 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:

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

In practice it might be more realistic to impose an acceptable range for the demand.

Multi-sourcing for multiattribute items can also require special consideration of *homogeneity constraints* when picking winners, such 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. In order to capture such a requirement we can 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:

$$z_{jk} \leq \sum_{i \in T_{jk}} x_{ijk} \leq |T_{jk}| \; z_{jk} \quad \forall j, k$$

$$0 \leq \sum_{k} z_{jk} \leq 1 \quad \forall j$$

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

4. Interactions between Computation and Incentives

Up to this point we have considered the computational complexity of exact implementations of economic mechanisms for market-based allocation problems. However, sometimes there is no reasonable implementation of the exact mechanism. In such cases, computational considerations must be introduced explicitly during the mechanism design process itself. This will be the focus of this section.

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.

4.1 Strategic Complexity

The *strategic complexity* 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 5.6) are proposed that make small deviations away from truthfulness unbeneficial to agents [PKE01], and the mechanism design problem has been considered with respect to an ϵ-strategyproofness [Sch02].

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].

4.2 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 2.4), 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. Holzman *et al.* [HKDMT01] describe necessary and sufficient conditions on the structure of bundles in the language to maintain strategyproofness[9] 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].

[9] 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.

4.3 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 [San00], 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, Par03]. 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 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 [CJ00], and derive necessary and sufficient conditions on information about agent preferences to be able to compute the VCG outcome in a combinatorial auction [Par02].

4.4 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, 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. 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 VCG mechanism is intractable. For example, Lehmann et al. [LOS02] propose a truthful and feasible mechanism for a combinatorial 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. Bartal *et al.* [BGN03] have proposed a truthful and feasible mechanism for the multi-unit combinatorial allocation problem, in which each bidder is restricted to demand a small fraction of the available units of each good.

One can also try to *distribute* the 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* [FS02], 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]. 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 [Sp03a]. This extends the consideration of truthful information revelation, present in classic mechanism design, to also require incentives for truthful information processing and computation [SP03b].

We can also consider relaxed strategic models, in which the goal of complete incentive-compatibility is relaxed. We briefly outline a taxonomy of strategic relaxations [FS02], and provide some examples of their use in the literature.

almost-strategyproofness *Approximate the strategic properties of a mechanism, perhaps along with other goals.* A particular example is the concept of ϵ-strategyproofness, in which an agent can gain at most ϵ by following some non-truthful strategy whatever the strategies of other agents. ϵ-strategyproofness has been considered by Kothari *et al.* [KPS03] for a multi-unit auction problem and Archer *et al.* [APTT03] for a setting of *known* single-minded bidders in which the auctioneer knows the bundles demanded by agents and only the value is private to each bidder.

tolerable manipulability *The kinds of manipulations are well characterized, and have tolerable effects on overall design goals.* This concept has been considered by Archer et al. [AT02], in a multicast cost-sharing setting.

feasible strategyproofness *There are beneficial manipulations available to agents, but they cannot compute them because of limited computational power.* Nisan & Ronen [NR00] have explored this in the context of the VCG mechanism, where they consider the equilibrium behavior of computationally limited agents within the context of approximate implementations of the VCG mechanism.

5. 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 also implement the outcome of the VCG mechanism (or variations), which reinforces the importance of Groves mechanisms in the design of practical mechanisms.

5.1 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 [DHK02], 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. Indeed, 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 combinatorial design for FCC auction #31,[10] and the Chicago GSB auction [GSS93], provide *linear-price* feedback along with the provisional allocation. Auctions such as *i*Bundle [PU00a] and A*k*BA [WW00] provide *non-linear price* feedback along with the provisional allocation. Other proposals,

[10] The Federal Communications Commission maintains a URL with documents and discussion on the design of auction #31 at http://wireless.fcc.gov/auctions/31/releases.

such as AUSM and an ascending-proxy design [AM02] are described without explicit price feedback.

Early theoretical results exist for particular restrictions on agent valuations. For example, the DGS auction [DGS86] solves the unit-demand (or *assignment*) problem, in which each agent wants at most one item. Another special case of a combinatorial auction for multiple identical items and decreasing marginal valuations is solved with Ausubel's auction [Aus02] (see Section 5.2.2).

More recently, Parkes & Ungar [PU00a] proposed an ascending auction, *i*Bundle, which is efficient for the case of *buyer-submodular* preferences. Buyer-submodular is slightly stronger than agents-are-substitutes.[11] As before, let $w(L)$ denote the coalitional value for agents $L \subseteq \mathcal{I}$. Buyer-submodular requires:

$$w(L) - w(L \setminus K) \geq \sum_{i \in K} \left[w(L) - w(L \setminus i) \right], \quad \forall K \subset L, \forall L \subseteq \mathcal{I}$$

Straightforward MBR bidding is an *ex post* Nash equilibrium in this case, with each agent choosing to bid in each round for the set of bundles that maximize its payoff given the current prices. This is proved by Ausubel & Milgrom [AM02] in their analysis of the closely-related ascending-proxy auction.

*i*Bundle proceeds in rounds. In each round agents can submit XOR bids on multiple bundles, at or above the current ask price. The auctioneer maintains non-linear, and perhaps non-anonymous ask prices. The auctioneer collects bids and solves a winner-determination problem, computing a provisional allocation to maximize revenue given the bids. Finally, prices are updated based on the bids from losing agents. The auction terminates when there are no losing bidders still active in the auction.

*i*Bundle Extend & Adjust (*i*BEA) [PU02] uses the concept of *Universal* competitive equilibrium prices (see Section 2.4) to extend *i*Bundle beyond its normal termination round and collect just enough additional information about agent preferences to compute VCG payments. Although the final payments by agents are computed as a discount from the final prices in the auction, *i*BEA is best viewed as an ascending-price auction because agents face increasing prices while actively bidding in the auction. The discount to agents is computed at the end of the auction, based on the final ask prices and following eq. (1.10).

Ausubel & Milgrom [AM02] have described a proxy-agent variation on *i*Bundle, in which bidders must submit preferences to proxy agents that sub-

[11] A simple extension, *i*Bundle & Adjust [PU00b, Par01], brings MBR into equilibrium for the agents-are-substitutes case.

mit ascending bids to an auction. The auction terminates with the VCG payments for the case of buyer-submodular values, and straightforward MBR is in an *ex post* Nash equilibrium for that case (just as in *i*Bundle). However, the ascending-proxy auction will in general terminate with agent payments that are greater than their payments in the VCG outcome, at least when agents choose to reveal their true values to the proxy agents. The ascending-proxy auction satisfies a *bidder-monotonicity property* that is not satisfied by the VCG. Bidder monotonicity states that revenue to the auctioneer increases with the number of agents and provides robustness against collusive bidding strategies. One potential drawback of the ascending-proxy auction, in comparison with a VCG-based mechanism such as *i*BEA, is that there are many equilibrium in ascending-proxy when buyer-submodular does not hold and agents must solve an implicit bargaining problem to implement equilibrium outcomes.

Many iterative combinatorial auctions, including the DGS auction [DGS86] and the *i*Bundle and *i*BEA auctions can be interpreted within the primal-dual design methodology described in Section 2.4. The following three steps are important in extending the primal-dual framework described in Section 2.4 to the combinatorial allocation problem:

a) choose an extended formulation of the efficient allocation problem so that the optimal LP relaxation is integral and so that the dual problem can be interpreted as defining the space of competitive equilibrium prices

b) implement a primal-dual algorithm for the efficient allocation problem, using myopic best-response information from agents in each round to adjust prices and the provisional allocation

c) terminate with *Universal* CE prices, to enable the computation of VCG payments and bring myopic best-response into equilibrium.

Step a) can be answered by appealing to the hierarchy of formulations for CAP provided by Bikhchandani & Ostroy [BO02b], each of which strengthens the formulation and has the effect of enriching the price space (to include both non-linear and non-anonymous prices). In fact, *i*Bundle tries to introduce non-anonymous prices only when necessary and can be thought of as implementing a primal-dual algorithm for a hybrid formulation for the CAP.

Step b) can be answered by paying careful attention to an economic interpretation of the complementary-slackness (CS) conditions. The goal is to demonstrate that the auction terminates with an allocation and prices that satisfy the CS conditions. We provide an outline of the proof of the properties of *i*Bundle, focusing on the special case (*i*Bundle(2)) that prices are still *anonymous*. See Parkes & Ungar [PU00a] for additional details. Let $x_i(S) = 1$ denote that bundle S is allocated to agent i, and let π_i denote the *maximal payoff* to agent

i at prices $p(S)$. The first important CS condition is:

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

In words, this condition states that any bundle allocated to an agent in the provisional allocation in a particular round should maximize its payoff at the prices. This is maintained (to within 2ϵ) in *i*Bundle when agents follow MBR strategies because the provisional allocation can only include bundles that receive bids from agents.

Let $y(k) = 1$ denote that partition $k \in \Gamma$ in the space of all feasible partitions is selected by the auctioneer. Partition $k \in \Gamma$ divides goods into bundles, but does not specify an agent assignment for the bundles. Notation $S \in k$ indicates that bundle S is in the partition. Let Π denote the maximal possible revenue to the auctioneer in the current round across all feasible allocations at the current prices (i.e. irrespective of the bids submitted by agents). The second important CS condition is:

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

In words, this condition states that the allocation must maximize the auctioneer's revenue at prices $p(S)$. Recall that the provisional allocation is selected to maximize revenue *given bids*, so it is necessary to show that the provisional allocation is maximal across all possible allocations despite this restriction. An important concept, known as *safety* is used to maintain this condition will hold (within $\min\{M, N\}\epsilon$) in all rounds of the auction. Essentially, safety ensures that the potential maximal increase in revenue due to price increases across rounds will always be matched by new (or renewed) bids from agents. Safety fails when the *exclusive-or* bids from a losing bidder are disjoint, at which point non-anonymous prices are introduced for that bidder.

Step c) can be achieved by continuing to adjust prices beyond the first round in which CE prices are determined for CAP($\mathcal{I}$). Instead, *i*BEA continues to adjust prices until the prices are also CE prices for all subproblems without each agent in the efficient allocation. Agents are kept ignorant of this second phase, and their bids serve to provide information that eventually discounts the final payments of other agents in order to implement the VCG payments.

5.2 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. This is a special-case of the combinatorial auction problem, in which the items are identical. Useful auction designs for this problem do not simply introduce an identifier for each item and use combinatorial auctions. Rather, a useful auction allows agents to submit bids and receive price feedback expressed in terms of the quantity of units of the item.

The *efficient* multi-unit auction problem has received some attention. We consider the forward auction problem, with one seller and multiple buyers, and distinguish between two simple cases: *unit-demand* valuations and *marginal-decreasing* valuations. Both cases can be solved with iterative auctions that maintain a *single* unit price. The first case is quite straightforward, but illustrative of the primal-dual auction design method. The second case requires an innovative "clinching" auction, in which price discounts are maintained for agents during the auction [Aus02]. This latter auction also has a direct interpretation as a primal-dual implementation for an appropriate LP formulation of the allocation problem [BdVSV01].

5.2.1 Unit Demand. In the unit demand setting each buyer demands a single item of the good and there are K items for sale. Let v_i denote agent i's value for one 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 single-item auction in Section 2.4, the VCG outcome is 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.

5.2.2 Marginal-Decreasing Values. With marginal-decreasing values, each buyer has a decreasing value for each additional unit that it receives in the assignment. We describe Ausubel's [Aus02] efficient ascending-price auction. There is no single price that supports the efficient allocation in competitive equilibrium for this marginal-decreasing case. However, the auction is able to maintain a single price along with enough additional information to compute discounts to each agent at the end of the auction, such that the final price faced by each agent is its VCG payment. This brings a simple MBR strategy into an *ex post* Nash equilibrium. The discounts are completely analogous to the discounts in *i*BEA.

Ausubel's 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 particular 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 that the demand from other agents will only fall as the price continues to increase. In essence, the future increase in ask price before the auction terminates represents the *discount* awarded to the agent for that item.

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. Marginal-decreasing Multi-unit Example.

Consider the simple example illustrated in Table 1.1, taken from Ausubel [Aus02]. 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.

5.3 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.

The single-item multiattribute auction, with single-sourcing, has received the most attention in the literature. In the efficient multiattribute auction problem the goal is to find the configuration of the item and seller that maximizes the total payoff across all participants. In the optimal multiattribute auction problem the goal is to find the configuration of the item, and a price, that maximizes the payoff to the buyer across all possible auction mechanisms. We retain the notation introduced in Section 3.2.4.

5.3.1 Efficient Multiattribute Auctions. In general, there can be *no* efficient multiattribute auction that does not run at a deficit. This follows from the Myerson-Satterthwaite impossibility result (see Section 2.2) because there is private information on two-sides of the market (for the buyer and for the sellers). In particular, the VCG mechanism runs at a deficit.

Instead, it is standard to consider a one-sided refinement to the VCG mechanism, that is budget-balanced but only approximately-efficient. In the one-sided VCG mechanism the allocation is computed as in the VCG mechanism. The only difference is that VCG payments are only provided on the sell-side, while the buyer is expected to make a payment equal to the VCG payment of the winning seller. Essentially, this leaves the buyer and the winning seller engaged in a *bargaining game* for a division of the total surplus $V(\mathcal{I})$. If the buyer reports a truthful cost function then the payoff division between the buyer and the winning seller is $[w(\mathcal{I} \setminus i^*), w(\mathcal{I}) - w(\mathcal{I} \setminus i^*)]$, where $w(L)$ is the maximal surplus with sellers $L \subseteq \mathcal{I}$ and i^* denotes the winning seller. A well-informed buyer can misstate her cost function and extract all the available surplus, to provide a payoff division of $[w(\mathcal{I}), 0]$. This difference in payoff, $w(\mathcal{I}) - w(\mathcal{I} \setminus i^*)$, places a simple and useful bound on the maximal benefit available to the buyer from deviating from a truthful strategy. As the auction becomes more competitive, the marginal product of any single seller becomes negligible and this maximal benefit tends to zero.

Parkes & Kalagnanam [PK03] have proposed a descending-price auction to implement the outcome of the modified VCG mechanism with MBR strategies. As with combinatorial auctions, the main advantage claimed for these price-based auctions is that they allow for incremental revelation of valuation and cost information, with price feedback guiding that process. MBR is an *ex post* Nash equilibrium strategy for the sellers against a *non-adaptive* (but perhaps untruthful) buyer strategy, and the same bounds on possible gains from manipulation are available for the buyer as in the one-sided VCG.

Of particular interest is auction ADDITIVE& DISCRETE (AD), which is designed for the special-case of preferential independence. Auction AD employs a compact price space, with linear-additive prices on attribute levels and an additional non-linear price term that applies to the bids from all suppliers. Experimental results for a simulated environment demonstrate that this price space provides a significant reduction in information revelation over an auction in which the price space maintains explicit prices on *bundles* of attribute levels.

Vulkan & Jennings [VJ00] also describe an iterative multiattribute auction design for this problem. However, their auction is not price based and the buyer– but not the seller –must reveal her complete valuation function to the auction.

5.3.2 Optimal Multiattribute Auctions.

Che [Che93] has proposed optimal multiattribute auctions for the special case of seller cost functions that are defined in terms of a *single* unknown parameter. Che's auctions are direct-revelation mechanisms, and he considers both first-price and second-price variations. The second-price variation is exactly the one-sided VCG mechanism, and Che is able to derive the optimal "scoring function" (or reported cost function) that the buyer should state to maximize her payoff in equilibrium. Branco [Bra97] extends the analysis to consider the case of correlated seller cost functions. No optimal multiattribute auction is known for a more general formulation of the problem, for example for the case of preferential-independence.

Recently, Beil & Wein [BW01] have proposed an *iterative* payoff-maximizing auction procedure, again for a class of parameterized cost and valuation functions with known functional forms and *naive* suppliers. The buyer restarts an auction a number of times to estimate the seller costs functions deterministically, restarting the auction with a different scoring function each time. For the final round, Beil & Wein design a scoring function that maximizes the buyer payoff by essentially reporting the same score (within ϵ) for the top two suppliers. They allow the buyer scoring function to change across rounds. The main assumption made about seller behavior is that a seller will continue to follow a straightforward bidding strategy, even though the auction is restarted many times and even though the information is used to learn their cost functions and finally extract the maximal possible payoff from the winning seller. This is clearly not a realistic equilibrium proposition.

5.4 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 maintenance, repair and operations services. 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.* [DHK02] have studied the use of reverse multi-unit auctions with volume discounts in a procurement setting. Winner determination formulations are provided in Section 3.2.2, where we also discuss the introduction of business rules as side constraints, which is an important consideration in practical electronic markets. A combinatorial auction is used for single units (lot) of multiple items, with all-or-nothing bids are allowed and nonlinear prices are used to feedback information. Volume discount auctions are also used, when multiple units of multiple items are being procured and for the restricted case of bids that are separable across items. 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. The competitive equilibrium property can be used to operationalize this notion of achieving a win-win outcome [DHK02].

The main technical challenge in analyzing the properties of such procurement reverse auctions is that the allocation problems 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 then an appeal can be made once again to primal-dual algorithms to show competitive equilibrium, as in the case of forward combinatorial auctions (see Section 5.1).

The reverse combinatorial auction is fundamentally different from the forward auction because: (i) set covering formulation rather than set packing, and (ii) the inclusion of business rules as side constraints that complicate the formulation. However, extended formulations can still be derived for the reverse combinatorial auction with side constraints. This extended formulation yields dual prices on bundles of items, just in the case of the forward auction.

In this section we outline an extended formulation that can be used to provide a primal-dual based iterative descending price auction for a procurement setting. Consider an extended formulation of the allocation problem corresponding to the set covering problem (1.11). The extended formulation is defined over the space of all feasible partitions of items with agent assignments, denoted Γ_{feas}. This time, we use $k \in \Gamma_{\text{feas}}$ to denote a partitioning of items into bundles *and* an assignment to agents. Let $y(k) = 1$ iff allocation k is selected, with $k \ni (S, i)$ to denote that the allocation assigns bundle S to agent i. $v_i(S)$ is the *cost* of allocating the bundle S to agent i, and variable $x_i(S)$ takes the value 1 if the bundle S is allocated to agent i. Note that the set Γ_{feas} is chosen such that only feasible partition-assignments are allowed, respecting side-constraints in addition to item feasibility constraints. For example a partition-assignment k 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 $x_i(S)$ have been relaxed.

$$
\begin{array}{lll}
\min & \sum_{S \subseteq \mathcal{G}} \sum_{i} v_i(S) x_i(S) & \\
\text{s.t.} & & \\
 & x_i(S) \geq \sum_{k \in \Gamma_{\text{feas}}, k \ni (S,i)} y(k), & \forall i,\ \forall S \\
 & \sum_{S \subseteq \mathcal{G}} x_i(S) \geq 0 & \forall i \\
 & \sum_{k \in \Gamma_{\text{feas}}} y(k) \geq 1 & \\
 & x_i(S), y(k) \geq 0 &
\end{array}
$$

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 i. The second constraint ensures that the total allocation to agent i 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:

$$\begin{array}{ll} \max & \Pi \\ \text{s.t.} & \\ & p_i(S) + \pi_i \leq v_i(S) \quad \forall i,\ \forall S \\ & \Pi \leq \sum_{(S,i)\in k} p_i(S) \qquad \forall k \in \Gamma_{\text{feas}} \\ & \Pi, \pi_i, p_i(S) \geq 0 \end{array}$$

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

$$\pi_i = \max\left\{0, \max_{S\subseteq\mathcal{G}}(v_i(S) - p_i(S))\right\}$$

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

$$\Pi = \min_{k\in\Gamma_{\text{feas}}} \sum_{(S,i)\in k} p_i(S)$$

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.

$$x_i(S) > 0 \Rightarrow \pi_i + p_i(S) = v_i(S) \qquad \text{(CS-1b)}$$

$$y(k) > 0 \Rightarrow \pi_S = \sum_{(S,i)\in k} p_i(S) \qquad \text{(CS-2b)}$$

Notice that the extended formulation uses a variable for each partition-assignment pair thereby introducing price discrimination across agents. Now a descending price auction with prices on each bundle can be used to reach the competitive equilibria following a primal-dual type algorithm [DHK02]. The bundle prices in each round converge towards a competitive equilibria when the suppliers following a straightforward bidding strategy.

5.5 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 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 at least δ can be done in polynomial time. This implies that in designs where the bids are not common knowledge, then each supplier and the buyer can engage in an *algorithmic protocol* to identify such proposals in polynomial number of steps. In addition, they show that such a mechanism converges to an competitive equilibrium solution where all suppliers are at their profit maximizing solution given the cost and the required cost decrement δ.

5.6 Double Auctions and Exchanges

Finally, it is interesting to consider the design of double auctions and exchanges in which their are multiple buyers and multiple sellers, present simultaneously in a market. As with multiattribute auctions, the problem that immediately arises is one of the economic impossibility of efficiency with budget-balance (by Myerson-Satterthwaite), again because both the buyers and sellers have private information about their preference structure.

In this section, we discuss settings with multi-unit homogenous item and multiple buyers and sellers, and combinatorial exchanges, with multiple het-

Name	traded	p_{buy}	p_{ask}	(EFF)	(BB)	equil	(IC)
VCG	l	$\max(A_l, B_{l+1})$	$\min(A_{l+1}, B_l)$	Yes	No	dom	yes
k-DA	l	$kA_l + (1-k)B_l$	$kA_l + (1-k)B_l$	No	Yes	BNE	no
TR-DA	$l-1$	B_l	A_l	No	Yes	dom	yes
McAfee-DA	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	dom	yes

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

erogeneous items and bids on bundles of items. Our examples are all from the *call market* paradigm in which bids and asks are collected and then cleared periodically. We ignore any temporal aspects to the problem.

5.6.1 Double Auctions. In a *double auction* there are multiple buyers and sellers, all interested in trading multiple units of the same item. 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 the 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 VCG 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 provides a summary of some of the double auction mechanisms (DAs) known in the literature. In terms of high-level properties of budget-balanced DAs, the following two characteristics are mutually-exclusive:

a) clearing the double auction to maximize the surplus given the bids and asks of agents

b) achieving incentive-compatibility, such that truthful bidding is an equilibrium strategy.

Properties a) and b) together would provide for an efficient and balanced DA, and violate the Myerson-Satterthwaite result. Notice that the TR-DA [BN01] and McAfee-DA [McA92] are truthful (in a dominant strategy), but deliberately clear the exchange to implement an inefficient trade. In comparison, the k-DA auction [Wil85, CS83, SW89] clears to maximize reported surplus but is not incentive-compatible. Of course, we know that the VCG mechanism is efficient but not balanced. The parameter $k \in [0, 1]$ in k-DA is chosen before the auction begins, with the clearing price faced by all agents calculated as $kA_l + (1-k)B_l$.

The McAfee DA 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. The TR-DA auction implements the fall-back option of McAfee's DA. An additional DA, the α-reduction DA [BN01], uses the TR-DA rule with probability α, and the VCG rule with probability $1-\alpha$. Parameter α can be chosen to make the mechanism *ex ante* balanced, and the α-reduction DA retains strategyproofness by mixing two strategyproof DAs.

5.6.2 Combinatorial Exchanges. Parkes *et al.* [PKE01] have suggested a family of VCG-based exchanges in which the exchange is cleared to implement the trade that maximizes reported value (or surplus). Naturally, the exchange is not incentive-compatible. 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 surplus 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)

Distance Function	Name	Definition	Parameter Selection
$\mathbf{L}_2, \mathbf{L}_\infty$	Threshold	$\max(0, \Delta_{\texttt{vick},i} - C_t^*)$	$\min C_t$ s.t. (BB')
$\mathbf{L}_{\mathrm{RE}}$	Small	$\Delta_{\texttt{vick},i}$, if $\Delta_{\texttt{vick},i} < C_s^*$ 0 otherwise	$\max C_s$ s.t. (BB')
$\mathbf{L}_{\mathrm{RE2}}$	Fractional	$\mu^* \Delta_{\texttt{vick},i}$	$\mu^* = V^* / \sum_i \Delta_{\texttt{vick},i}$
$\mathbf{L}_{\mathrm{WE}}$	Large	$\Delta_{\texttt{vick},i}$, if $\Delta_{\texttt{vick},i} > C_l^*$ 0 otherwise	$\min C_l$ s.t. (BB')
$\mathbf{L}_\Pi$	Reverse	$\min(\Delta_{\texttt{vick},i}, C_r^*)$	$\max C_r$ 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 $|I^*|$ (used in the Equal rule) is the number of agents that participate in the trade.

ensure that no agent receives more than its Vickrey discount. The authors consider a variety of distance functions, including standard metrics such as $\mathbf{L}_2(\boldsymbol{\Delta}, \boldsymbol{\Delta}_{\texttt{vick}}) = \sum_i (\Delta_{\texttt{vick},i} - \Delta_i)^2$ and $\mathbf{L}_\infty(\boldsymbol{\Delta}, \boldsymbol{\Delta}_{\texttt{vick}}) = \max_i (\Delta_{\texttt{vick},i} - \Delta_i)$. 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, $\Delta_i^*(C_t) = \max(0, \Delta_{\texttt{vick},i} - C_t)$, 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. These results were first reported by Parkes *et al.* [PKE01], with the experimental results computed through a simple approximation to a restricted Bayesian Nash equilibrium. The experimental results have since received additional validation by Krych's analysis [Kry03], in which an exact Bayesian-Nash equilibrium is computed, although for a very restricted set of agent strategies. 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, which points to the face that an agent's discount in Large is very sensitive to its bid and suggests that Large is likely to be less robust than Threshold in practice.

6. Discussion

Our goal in this chapter has been to provide a cohesive overview of the different auction mechanisms that are commonly encountered in practice and in the literature. We have emphasized a theoretical approach, and we have taken a cue from traditional mechanism design (for example via an appeal to the revelation principle) in order to understand the problem of designing incentives to solve optimization problems in settings with rational agents. We have often introduced computational techniques, for example via carefully structured bidding languages and primal-dual algorithms, in order to also achieve useful computational properties.

In closing, we find it useful to take a step back and consider the role of theoretical MD in the design of auctions and exchanges for electronic markets. Certainly, mechanism design is a powerful tool which has produced some very interesting results, both positive and negative. 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. Thus, MD provides the designer with a useful "model and optimize" mindset. 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 can be used to construct indirect implementations of VCG-based mechanisms, there are no general methodologies to develop indirect implementations for other classes 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 the inclusion of incentive-compatibility constraints in a model.

It seems possible that computational methods can be used to begin to address the first problem, that of the difficulty of the mechanism design problem.

Conitzer & Sandholm [CS02b] propose *automated mechanism design*, in which a computational method is used to design mechanisms with respect to highly-enumerative description of the function space and agent type space. The challenge in this automated MD approach is to develop structured representations of the problem to constrain the input size to the optimization. However, automated MD cannot solve the wider issues presented by the second two problems because it is only applicable to direct revelation mechanisms and because it continues to ignore computational considerations in the formulation of the problem.

Experimental mechanism design (e.g. [PMPS02]) presents an alternative paradigm, in which computational methods are used to determine the performance of a particular mechanism design. The design problem can then be formulated as a search through a structured space, itself chosen to be expressive enough to capture a class of interesting mechanisms. Experimental MD remains within the spirit of classic mechanism design because it continues to seek to maximize performance with respect to beliefs about the way that self-interested agents will participate. However, the analysis of a mechanism is done through explicit computation instead of the imposition of incentive-compatibility constraints at design time.

The methodology of experimental MD mirrors that of experimental economics, in which experiments with human subjects in carefully controlled laboratory settings are used to test theoretical predictions and to assess the robustness of a mechanism to unmodeled behaviors [Mil02, chapter 1]. There is a growing recognition of the importance of experimental methodologies within the study of human economies. Indeed, Al Roth, an economist involved in the design of real-world markets such as those used in the medical resident matching program has recently advocated an "economics as engineering" approach [Rot02] to market design. It seems interesting to turn to computational methods to test and design mechanisms that will be used by computational trading agents.

Experimental MD offers a number of benefits over both analytic MD and automated MD. First, it extends to indirect mechanisms, because the design space is limited only to mechanisms for which equilibrium behavior can be computed through computational methods. Second, considerations of agent bounded-rationality and off-equilibrium play can also be considered explicitly within the model, again because computational methods are used to design and evaluate the mechanism. In particular, we are not limited to incentive-compatible design in the framework of experimental MD. However, experimental MD presents three main challenges. Briefly, the challenges (and some initial directions) are those of:

evaluation Take a mechanism description and compute the performance of the mechanism with respect to models of agent self-interest and rationality. Current approaches proposed for this problem include using *genetic programming* primitives to evolve agent trading strategies [PMPS02], and methods to compute a restricted Bayesian-Nash equilibrium, including *replicator dynamics* [WGSW02], *fictitious play* [ZW02], and *best-response dynamics* [Kry03, RW03].

optimization Implement the mechanism optimizer, which searches in the space of the mechanism description language for a good mechanism. One preliminary approach proposed for this problem is to allow *genetic programming* to evolve the rules of a parameterized family of mechanisms [PMPS02, Cli01].

description Define a mechanism description language, to provide the interface between the mechanism optimizer and the black box mechanism evaluator. Wurman *et al.* [WWW02] have proposed a parameterization of the auction design space, and there has been some work to develop a declarative approach to the specification of negotiation structures and auction rules [RWG01] and to develop ontologies for automated negotiation [TWD02].

Although each of these problems has received some attention in isolation in recent years there certainly remain significant computational difficulties if progress is to be made in this agenda of experimental mechanism design. In the meantime it seems likely that continued progress will be made in introducing computational considerations into the mechanism design program, and in the constructive application of mechanism design theory to electronic market design.

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.

[AT02] A Archer and E Tardos. Frugal path mechanisms. In *Proc. 13th Symp. on Discrete Algorithms*, pages 991–999, 2002.

[APTT03] Aaron Archer, Christos Papadimitriou, Kunal Talwar, and Eva Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proc. 14th ACM Symposium on Discrete Algorithms*, 2003.

[Aus02] Lawrence M Ausubel. An efficient ascending-bid auction for multiple objects. *American Economic Review*, 2002. Forthcoming.

[AC99] Lawrence M Ausubel and Peter Cramton. The optimality of being efficient. Technical report, University of Maryland, 1999.

[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.

[BGN03] Y Bartal, R Gonen, and N Nisan. Incentive compatible multi unit combinatorial auctions. Technical report, The Hebrew University of Jerusalem, 2003.

[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.

[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. in Proceedings of ICS 2003.

[BdVSV01] Sushil Bikhchandani, Sven de Vries, James Schummer, and Rakesh V Vohra. Linear programming and Vickrey auctions. In Brenda Dietrich and Rakesh Vohra, editors, *Mathematics of the Internet: E-Auction and Markets*, pages 75–116. IMA Volumes in Mathematics and its Applications, Springer-Verlag, 2001.

[BO02a] Sushil Bikhchandani and Joseph M Ostroy. Ascending price Vickrey auctions. *Games and Economic Behavior*, 2002. Forthcoming.

[BO02b] Sushil Bikhchandani and Joseph M Ostroy. The package assignment model. *Journal of Economic Theory*, 2002. Forthcoming.

[BN02] Liad Blumrosen and Noam Nisan. Auctions with severely bounded communication. In *Proc. 43rd Annual Symposium on Foundations of Computer Science*, 2002.

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

[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.

[CP03] Estelle Cantillon and Martin Pesendorfer. Combination bidding in multi-unit auctions. Technical report, HBS and LSE, 2003.

[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.

[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, CERAS and UCL, 2000.

[CS01] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions. In *Proc. 3rd ACM Conf. on Electronic Commerce (EC-01)*, pages 256–259. 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. .

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

[DHK02] Andrew Davenport, Abigail Hohner, and Jayant Kalagnanam. 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.

[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, 2002.

[EGKL01] Marta Eso, Soumyadip Ghosh, Jayant R Kalagnanam, and Laszlo Ladanyi. Bid evaluation in procurement auctions with piecewise linear supply curves. Technical Report RC 22219, IBM TJ Watson Research Center, 2001.

[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.

[FS02] Joan Feigenbaum and Scott Shenker. Distributed Algorithmic Mechanism Design: Recent Results and Future Directions. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13, 2002.

[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.

[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.

[GSS93] Robert L Graves, Linus Schrage, and Jayaram Sankaran. An auction method for course registration. *Interfaces*, 23(5):81–92, 1993.

[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.

[HKDMT01] R Holzman, N Kfir-Dahav, D Monderer, and M Tennenholtz. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*, 2001.

[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 at AAMAS'02*, 2002.

[Jac00] Matthew O. Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.

[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.

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

[KPS03] Anshul Kothari, David C. Parkes, and Subhash Suri. Approximately-strategyproof and tractable multi-unit auctions. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, pages 166–175, 2003.

[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.

[Kry03] David Krych. Calculation and analysis of Nash equilibria of Vickrey-based payment rules for combinatorial exchanges. Technical report, Harvard University, 2003. Undergraduate Thesis. Available from http://www.eecs.harvard.edu/econcs.

[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. *Interfaces*, 2000. Forthcoming.

[LOS02] Daniel Lehmann, Liadan Ita O'Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, September 2002.

[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.

[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.

[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 *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, volume 2531 of *Lecture Notes in Artificial Intelligence*, pages 103–122. 2002.

[Par03] David C Parkes. Auction design with costly preference elicitation. Submitted for publication, 2003.

[PK03] David C Parkes and Jayant Kalagnanam. Iterative Multiattribute Vickrey Auctions. Submitted for publication, 2003.

[PKE01] David C Parkes, Jayant R Kalagnanam, and Marta Eso. Achieving budget-balance with Vickrey-based payment schemes in 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. Agent-Mediated Electronic Commerce (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.

[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.

[RW03] Daniel M. Reeves and Michael P. Wellman. Computing equilibrium strategies in infinite games of incomplete information. In *Proc. of Game-theoretic and Decision-theoretic Workshop at AAMAS'03*, 2003.

[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.

[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.

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

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

[San02] Tuomas W Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18:656–676, 2002.

[SS01] Tuomas W Sandholm and Subhash Suri. Market clearability. In *Proc. 17th 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.

[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.

[Sch02] James Schummer. Almost dominant strategy implementation. Technical report, MEDS Department, Kellogg Graduate School of Management, 2002.

[Sp03a] Jeff Shneidman and David C. parkes. Rationality and self-interest in peer to peer networks. In *2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003.

[SP03b] Jeffrey Shneidman and David C. Parkes. Using redundancy to improve robustness of distributed mechanism implementations. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, 2003. (Poster paper). Extended version at http://www.eecs.harvard.edu/˜parkes/pubs/redundancy.pdf.

[TWD02] Valentina Tamma, Michael Wooldridge, and Ian Dickinson. An ontology based approach to automated negotiation. In *Proc.*

Agent-Mediated Electronic Commerce (AMEC'IV) workshop at AAMAS'02, 2002.

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

[VJ00] N Vulkan and N R Jennings. Efficient mechanisms for the supply of services in multi-agent environments. *Decision Support Systems*, 28:5–19, 2000.

[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.

[WW00] Peter R Wurman and Michael P Wellman. A*k*BA: A progressive, anonymous-price combinatorial auction. In *Second ACM Conference on Electronic Commerce*, pages 21–29, 2000.

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

[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)*, pages 385–390, July 2002.