Computational Mechanism Design

Depth Examination Report

Nathanaël Hyafil

Department of Computer Science
University of Toronto

nhyafil@cs.toronto.edu

March 9, 2005

Contents

1 Introduction

With the emergence of the Internet as a global structure for communication and interaction, many "business to consumer" and "business to business" applications have migrated online, thus increasing the need for software agents that can act on behalf of people, institutions or companies with private and often conflicting interests. To deal with these conflicts, such agents, and the protocols through which they interact should be designed according to the well studied principles of game theory (Osborne and Rubinstein (1994)) and mechanism design (Mas-Colell *et al.* (1995)).

Mechanism design deals with the design of incentives for a distributed population of agents to behave in a way that will lead to an optimal global outcome. Auctions in particular have proven particularly useful in allocating items in an economically efficient way. Combinatorial auctions (Demange *et al.* (1986a); MacKie-Mason and Varian (1994)), for example, can be used for allocating resources (or tasks) to agents with general non-additive preferences that exhibit complementarity or substitutability. Multi-attribute auctions (Bichler and Kalagnanam (2002)) or exchanges allow agents to negotiate over non-price attributes such as quality, color, delivery time and so on, so as to improve economic efficiency in markets with configurable goods.

One of the central positive result of "classical" mechanism design is the family of mechanisms called Vickrey-Clarke-Groves (VCG). Much (although not all) of the work on VCG mechanisms in the economics literature has largely ignored a number of issues: the limited ability of users to accurately assess and express their own preferences, the cost of revealing these preferences to others, and the computational burden of executing a mechanism in large outcome spaces. These issues are central to the emerging field of computational mechanism design, at the intersection of game theory and computer science. Research in this area follows one of two approaches: the first is to maintain the focus on VCG mechanisms and attempt to address previously ignored issues, independently, within that framework. The second is to change the framework entirely and design a new class of mechanisms that do not suffer the same deficiencies yet achieve the same objectives.

The purpose of this document is to concisely review the major results in the computational mechanism design literature. We first quickly review the necessary background in game theory and classical mechanism design (Section 2), before identifying more precisely the complexities that are the central motivations for computational mechanism design (Section 3). Section 4 surveys methods that attempt to remove these complexities within the framework of VCG mechanisms, while Section 5 focuses on mechanisms of a different form, namely sequential mechanisms. In Section 6, we study the automation of the design of mechanisms tailored to specific settings, before proposing future research directions of particular interest.

Other applications of mechanism design, such as online mechanism design, where agents arrive and depart dynamically at uncertain times, or distributed mechanism design, where there is no central entity enforcing the rules of the mechanism, have also been considered part of computational mechanism design. They are, however, outside the scope of this document.

2 Background

In this section we provide the minimum background necessary before reviewing computational mechanism design. For a more comprehensive review of utility theory see (Keeney and Raiffa (1976)) and of game theory and mechanism design see (Osborne and Rubinstein (1994); Mas-Colell *et al.* (1995)).

2.1 Utility and Game Theory

2.1.1 Preferences and Utility functions

Since the purpose of game theory is to analyze or predict how agents with conflicting interests interact, we need a way to represent these interests, or preferences. A *preference relation* is defined as a binary relation over the set of possible outcomes: $x \succeq y$ is interpreted as "outcome x is at least as good as outcome y".

When facing uncertainty, preferences are defined over a set of lotteries, i.e., a specification of the probability that each outcome will occur (von Neumann and Morgenstern (1944)). Preference relations can only represent ordinal information about agents' preferences and in most settings this will not be sufficient. Instead, we would like to be able to give a cardinal value to each outcome (its utility) and lottery (its expected utility). It can be shown that under very reasonable assumptions on a preference relation $\succeq$, there exists a function u, linear and continuous in the outcome probabilities, such that for any two lotteries L and L', $L \succeq L'$ if and only if $u(L) \geq u(L')$. Such a u is said to *represent* $\succeq$ and is often called a von-Neumann Morgenstern utility function. Note that u is not unique: given a utility function representing $\succeq$, any positive affine transformation of it will also represent $\succeq$.

In many settings, a utility function is assumed to have some additional structure; in this document we will often consider quasi-linear utilities: a quasi-linear environment is one where the outcome space X can be divided into a non-numeraire part K (often called an allocation) and a payment for each of the agents, $\mathbb{R}^N$. In such an environment, agent i's utility function is said to be quasi-linear if for any outcome $x = (k, p_1, \ldots, p_N) \in X = K \times \mathbb{R}^N$ we have $u_i(x) = v_i(k) - p_i$. v_i is called agent i's valuation function, and p_i its payment.

Preference elicitation deals with the problem of learning enough information about an agent's preferences in order to make a decision on that agent's behalf. As we will see, this will often involve querying the agent in an iterative process. A commonly used type of query is the *standard gamble query*, where an agent is asked to choose between an outcome and a standard lottery, that is, a lottery between the best and the worst outcome only.[1] Other types of queries include value queries ("what is your utility for outcome x?"), rank queries ("what is your j^{th} preferred outcome?") or demand queries ("which outcome would you "buy" if these were the prices?").

2.1.2 Bayesian Games

Game theory studies many forms of games, but the type of games relevant to mechanism design is those where there is uncertainty about the agents' preferences: *Bayesian games*. The following definitions and notation have been adapted from Osborne and Rubinstein (1994); Mas-Colell *et al.* (1995).

Definition 1 *A* Bayesian Game *consists of:*

- *a set of N agents*
- *a product set of actions $S = S_1 \times \ldots \times S_N$, one set for each agent*
- *a product set of types $\Theta = \Theta_1 \times \ldots \times \Theta_N$, one set for each agent*
- *a commonly known joint prior on the agents' types: $\Phi(\Theta)$*
- *a utility function $u_i : S \times \Theta_i \to \mathbb{R}$, for each agent*

Unlike above, the utility is defined not only over outcomes or lotteries (induced by a joint action of the agents) but also over types. This is just a difference in representation: the utility mapping itself is assumed to be commonly known and the type of the agents represent the aspects of the utility function that are privately known by that agent.

In a Bayesian game, a strategy for an agent is a mapping from possible types to (possibly mixed) actions: $\sigma_i : \Theta_i \to \Delta(S_i)$. We will be interested in outcomes where the agents' strategies are in some form of equilibrium. There are three main equilibrium concepts: dominant strategy, ex-post, or Bayes-Nash equilibria.

Definition 2 σ_i^* *is a* dominant strategy *for agent i if and only if*
$\forall \theta_i \in \Theta_i, \forall \sigma_{-i} \in \Sigma_{-i}, \forall \theta_{-i} \in \Theta_{-i} \forall \sigma_i' \in \Sigma_i,$
$u_i(\sigma_i^*(\theta_i), \sigma_{-i}(\theta_{-i}), \theta_i) \geq u_i(\sigma_i'(\theta_i), \sigma_{-i}(\theta_{-i}), \theta_i)$

Intuitively, a dominant strategy maximizes an agent's utility regardless of the other agents' actions.

A dominant strategy equilibrium (DSE) $\sigma^* = (\sigma_1^*, \ldots, \sigma_N^*)$ is a strategy profile where for each i, σ_i^* is a dominant strategy.

Definition 3 $\sigma^* = (\sigma_1^*, \ldots, \sigma_N^*)$ *is an* ex-post equilibrium *if and only if*
$\forall \theta_i \in \Theta_i, \forall \sigma_i' \in \Sigma_i, \forall \theta_{-i} \in \Theta_{-i}$
$u_i(\sigma_i^*(\theta_i), \sigma_{-i}^*(\theta_{-i})), \theta_i) \geq u_i(\sigma_i'(\theta_i), \sigma_{-i}^*(\theta_{-i})), \theta_i)$

[1] A special case of this is the order query, which offers a choice between two (deterministic) outcomes.

Intuitively, in an ex-post equilibrium, each agent's strategy maximizes his utility against the fixed strategies of the other agents but regardless of their types.

Definition 4 $\sigma^* = (\sigma_1^*, \ldots, \sigma_N^*)$ *is a* Bayes-Nash equilibrium *(BNE) if and only if*
$\forall \theta_i \in \Theta_i, \forall \sigma_i' \in \Sigma_i,$
$E_{\theta_{-i}}[u_i(\sigma_i^*(\theta_i), \sigma_{-i}^*(\theta_{-i})), \theta_i)|\theta_i] \geq E_{\theta_{-i}}[u_i(\sigma_i'(\theta_i), \sigma_{-i}^*(\theta_{-i})), \theta_i)|\theta_i]$

Intuitively, in a BNE, an agent's strategy maximizes his *expected* utility against the fixed strategies of the other agents, with the expectation taken with respect to his beliefs about the others' types.

A dominant strategy equilibrium is an ex-post equilibrium, which itself is a Bayes-Nash equilibrium. The converses are obviously not true. A Bayes-Nash equilibrium, however, always exists whereas there is no such guarantee for the others.

2.2 Classical Mechanism Design

2.2.1 Basic Concepts

Mechanism design deals with the design of incentives for a distributed population of agents to behave in a way that will lead to an optimal global outcome. The meaning of "optimal" depends on the mechanism designer's interests. These are represented by a social choice function that maps any type profile of the agents to the outcome that the designer would like to see occur if that type profile is realized. The following definitions and notation are taken from (Mas-Colell *et al.* (1995)).

Definition 5 *A* social choice function *is a function* $f : \Theta_1 \times \ldots \times \Theta_N \to X$ *that assigns a collective choice* $f(\theta_1, \ldots, \theta_N) \in X$ *to each type profile* $(\theta_1, \ldots, \theta_N)$*.*

f is called *ex post efficient* if for no profile $\theta = (\theta_1, \ldots, \theta_N)$ is there an $x \in X$ such that $u_i(x, \theta_i) \geq u(f(\theta), \theta_i), \forall i$ and $u_i(x, \theta_i) > u(f(\theta), \theta_i)$, for some i. In other words, no change of outcome can increase one agent's utility without decreasing another agent's.

We now formally define a mechanism:

Definition 6 *A* mechanism $\Gamma = (S_1, \ldots, S_N, g)$ *consists of* N *sets of actions, one for each agent, and an outcome function* $g : S_1 \times \ldots \times S_N \to X$*.*

Intuitively, a mechanism defines the rules of a game by specifying which outcome will occur for every possible joint action of the agents. In fact, a mechanism, along with a utility function for each agent and a prior over these utilities, induces a Bayesian game.

Mechanism $\Gamma = (S_1, \ldots, S_N, g)$ is said to *implement* social choice function f *in dominant strategies* (respectively, *in Bayes-Nash equilibrium*) if there is a strategy profile $\sigma^* = (\sigma_1^*, \ldots, \sigma_N^*)$ that is a dominant strategy (respectively, Bayes-Nash) equilibrium of the Bayesian game induced by Γ and any set of utility functions, such that $\forall \theta \in \Theta, g(\sigma^*(\theta)) = f(\theta)$.

A special class of mechanisms, called *direct revelation mechanisms*, is of particular interest:

Definition 7 $\Gamma = (S_1, \ldots, S_N, g)$ *is a* direct revelation *mechanism if* $S_i = \Theta_i, \forall i$ *and* $g = f$.

That is, agents directly reveal their types and the mechanism maps type profiles to outcomes.

A social choice function f is said to be *truthfully implementable* in DSE (respectively, BNE) if there exists a DSE (resp., BNE) in the direct revelation mechanism $\Gamma = (\Theta_1, \ldots, \Theta_N, f)$, where all agents reveal truthfully (i.e., $\sigma_i(\theta_i) = \theta_i, \forall i$).

Truthful implementation in dominant strategies is more commonly called *incentive compatibility*, and in Bayes-Nash equilibrium *Bayesian incentive compatibility.*

A major result in mechanism design, the Revelation Principle, says that we can restrict ourselves to direct revelation mechanisms without loss of generality.

Theorem 1 Revelation Principle: *If* $\Gamma = (S_1, \ldots, S_N, g)$ *implements social choice function* f *in dominant strategies (resp., BNE) then* f *is truthfully-implementable in dominant strategies (resp., BNE).*

Intuitively, this is because the mechanism can be modified so as to simulate the non-truthful strategy of the agents, therefore making it in their best interest to reveal truthfully since the mechanism will "lie for them."

Another relevant property of a mechanism is *individual rationality* (IR), which requires that an agent always prefers to participate in the mechanism. *Ex-post IR* requires that agents choose participation even if they knew the types of the others, whereas for *ex-interim IR* agents are only assumed to know their own type.

2.2.2 Negative Results

One of the most important results in mechanism design is a negative one. It states that in the general case, the range of social choice functions that can be implemented in dominant strategies is very restricted. More formally:

Theorem 2 Gibbard-Satterthwaite Theorem: *When* $2 < |X| < \infty$ *and agents can have any rational strict preferences on* X*, and* $f(\Theta) = X$*, then:*
f *is truthfully-implementable in dominant strategies if and only if* f *is dictatorial, where a social choice function is said to be dictatorial if it selects the outcome according to the preferences of only one of the agents.*

There are however two ways to circumvent this negative result: we can relax the implementation concept to Bayesian incentive compatibility, or we can restrict the set of preferences that an agent can have. In the first case, we unfortunately have another negative result:

Theorem 3 Myerson-Satterthwaite Theorem: *In a bilateral trade of a good, where the seller's cost is in* $[\underline{c}, \overline{c}]$ *and the buyer's valuation in* $[\underline{v}, \overline{v}]$ *with* $\underline{c} \leq \overline{v}$*, there is no Bayesian incentive compatible social choice function that is both ex-post efficient and individually rational.*

2.2.3 A Positive Result

In the second case, if we restrict our attention to quasi-linear environments, we have a very important positive result: there is a family of mechanisms that truthfully implements a very interesting class of social choice function, namely those with an ex-post efficient allocation function. Formally:

Definition 8 *A* Groves Mechanism *is a direct revelation mechanism* $\Gamma = (\Theta_1, \ldots, \Theta_N, f)$ *with* $f = (k^*, p_1, \ldots, p_N)$ *and where:*

- $\forall k \in K, \sum_i v_i(k^*(\theta), \theta_i) \geq \sum_i v_i(k, \theta_i)$
- $\forall i \in N, p_i(\theta) = -\sum_{j \neq i} v_j(k^*(\theta), \theta_j) + h_i(\theta_{-i})$*, for an arbitrary function* h_i

A *Groves-Clarke* mechanism is a Groves mechanism with:
$h_i(\theta_{-i}) = \sum_{j \neq i} v_j(k^*_{-i}(\theta_{-i}), \theta_j)$ where k^*_{-i} satisfies
$\forall k \in K, \sum_{j \neq i} v_j(k^*_{-i}(\theta), \theta_i) \geq \sum_{j \neq i} v_j(k, \theta_i)$

In other words, a Groves mechanism has an ex-post efficient allocation function and each agent pays an amount independent of his own revealed type minus the social welfare of the others in the efficient allocation. In a Groves-Clarke mechanism, the arbitrary functions h_i represent the social welfare of the other agents in the allocation that would be efficient if agent i had not participated. Thus in a Groves-Clarke mechanism, agents pay what their presence costs to others.

Proposition 1 *A Groves mechanism* $\Gamma = (\Theta_1, \ldots, \Theta_N, f = (k^*, p_1, \ldots, p_N))$ *truthfully implements* f *in dominant strategies.*

Furthermore, if, for each agent, all valuation functions mapping K to $\mathbb{R}$ are possible given the type space, and if k^* is ex-post efficient, then Groves is the only class of mechanisms that truthfully-implements $f = (k^*, p_1, ..., p_N)$

2.2.4 Basic Auction Theory

The most common application of mechanism design is auction design. An *auction* is a mechanism to allocate a set of goods to a set of agents and determine the payments. The auction specifies which bids are allowed (the actions of the mechanism), and for all possible bids of the agents, who is assigned which goods and at what price (the outcome rule). We will always assume, in this survey, that agents have independent private values, meaning that an agent's utility for an outcome does not depend on the other agents' utilities (private) nor does it depend on the allocation and payments that other agents receive (independent).

There are two main categories of auctions: sealed-bid and open-cry. In a sealed-bid auction, agents' bids are only observable by the auctioneer whereas in an open-cry auction they are public. For the allocation of a single good, there are four main auctions, two of each category. In a First-Price Sealed-Bid auction, the agent who submitted the highest bid wins the item and pays the amount that he bid, whereas in a Second-Price Sealed-Bid that same agent wins and pays the amount of the second highest bid. In an English, or Open-Ascending Price, auction the auctioneer announces an ask price, and agents must decide if they want to stay in the auction or drop out; the auctioneer increases the ask price incrementally until only one agent remains. A Dutch, or Open-Descending, auction is similar except that the initial ask price is high and the auctioneer incrementally decreases it.

Strategically, the Second-Price and English auctions are equivalent (under the private values assumption): it is a dominant strategy to bid truthfully by bidding your exact valuation in the former and staying active as long as the ask price is below your valuation in the latter. Intuitively, this is because the payment of the winner in these auctions (in both cases the second highest valuation) is independent of the winner's bid. The First-Price and Dutch auctions are also equivalent but there is no dominant strategy equilibrium here. There is a Bayes-Nash equilibrium (as always) which consists of bidding the expected highest valuation of the others assuming that it is below one's true valuation. In both cases the auction is efficient (the agent with the highest valuation wins) but only the Second-Price and English auctions are incentive compatible (truth-telling is dominant).

Intuitively it might seem that, because of their definitions, the First-Price auction provides the seller with more revenue than a Second-Price one. However, a famous result in auction theory, the Revenue Equivalence Principle, states that if agents' valuations are independently and identically distributed (IID), any two auctions that satisfy some basic assumptions lead to the same amount of revenue for the seller. Thus the First- and Second-Price auctions are also revenue-equivalent. They are not, however, revenue-optimal: as we will see in Section 6, for any given prior, the corresponding Myerson auction maximizes revenue.

2.2.5 Combinatorial and Multi-Attribute Auctions

Most (although not all) of the focus of *computational* mechanism design will be on auctions with more complex outcome spaces than the standard one-item auctions introduced above, in particular *multi-attribute* (MAA, Bichler and Kalagnanam (2002)) and *combinatorial* (CA, Demange *et al.* (1986a); MacKie-Mason and Varian (1994)) auctions.

In the basic multi-attribute setting there is still only one object being auctioned but some of the properties (attributes) of that object are part of the negotiation. For example, consider the case of a traveler looking to book a hotel room in city C. There are several possible hotels and each hotel offers rooms with varying attributes (Cable TV, Wifi Internet, room/suite etc...). The traveler will have a certain value for each possible configuration of a hotel room (it is usually assumed that the value depends only on the good and not on the identity of the seller, here the hotel, but this can be relaxed) and each hotel incurs a certain cost for such a configuration. The goal of the (here, reverse) auction is to determine which good will be traded (the room configuration), from which seller (the hotel) and at what price, so as to optimize some objective function, typically maximizing the surplus (buyer's value minus seller's cost, which is equal to social welfare) or the auctioneer's utility (e.g., revenue). The outcome space is therefore decomposed as $X = K_1 \times \ldots \times K_p \times N \times \mathbb{R}^N$, where N is the set of sellers, there are p attributes and attribute j takes values in K_j.

This traveler also needs to find transportation to city C (say, by plane) and might also want to book some entertainment events during his stay (e.g., theater tickets, sports games...). This is the combinatorial setting, where several items are being auctioned and these items' values are not independent. For example, the value of a hotel room or a baseball game on a specific day will obviously depend on whether or not the plane tickets allow the person to be in town that day. Also the value of an entertainment event could depend on the distance between its venue and the hotel. In this setting, the auction's goal is to determine which group of items to buy, from which seller and at what price, so as to optimize the objective, typically social welfare or the auctioneer's utility.[2] If K is the set of all items for sale, the outcome space is $X = N^K \times \mathbb{R}^N$, where N^K is the set of allocations of goods to sellers and $\mathbb{R}^N$ the prices paid to each seller. Unlike in the basic multi-attribute setting above, there are several objects being sold so their can be several different sellers. The literature usually treats MAAs and CAs separately although one can imagine the general case of exchanging multiple items, each with multiple attributes, between multiple buyers and sellers.

Most of the literature on MAAs and CAs focuses on *efficient* auctions, that is where the objective is to maximize social welfare (as opposed to *optimal* auctions which attempt to maximize the auctioneer's utility). But social welfare does not depend on any monetary transfer between agents, it "only" depends on the non-numeraire part of the selected outcome and therefore on the allocation function k. We have seen that the Groves family of mechanisms truthfully implements (in dominant strategies) any social choice function whose allocation function is efficient. When all possible valuation functions arise for some type of the agent Groves mechanism are actually the only mechanisms with that property. Not surprisingly, Groves mechanisms are a central part of computational mechanism design, especially the special case of Groves-Clarke mechanisms, more commonly referred to as Vickrey-Clarke-Groves (VCG)

[2]For consistency, this example uses a *reverse* combinatorial auction. Typically the literature deals with the case of one seller selling to several buyers.

mechanisms, or in the CA and MAA context as the Generalized Vickrey Auction (GVA).

In a combinatorial auction, social welfare is simply the sum of the buyers' valuations for the outcome ($SW(k,\theta) = \sum_{i=1}^{N} v_i(k(\theta),\theta_i)$), while in a multi-attribute setting it is the surplus, that is the difference between the buyer's valuation and the winning seller's cost ($SW((k,i),\theta) = v_B(k(\theta),\theta_B) - c_{i(\theta)}(k(\theta),\theta_{i(\theta)})$). The GVA therefore works as follows:

1. agents submit their valuation or cost function directly to the auction manager

2. the non-numeraire part of the outcome is selected so as to maximize social welfare: $\forall\theta$,

$$k^*(\theta) \in arg \max_{k(\cdot)} SW(k,\theta), \text{ in a CA or} \tag{1}$$

$$(k^*(\theta), i^*(\theta)) \in arg \max_{k(\cdot,i(\cdot))} SW((k,i),\theta), \text{ in an MAA} \tag{2}$$

3. the payments are determined by $p_i(\theta) = SW_{-i}(k^*,\theta) - SW(\hat{k}^{-i},\theta_{-i})$, where SW_{-i} is the social welfare of all agents except i and $\hat{k}^{-i}$ is the efficient allocation function in the auction where agent i is removed from the participants.

In both settings the GVA is dominant incentive compatible and ex-post individually rational (by construction of the payments). There are however a number of practical issues with the GVA, which we discuss further in the next section.

3 Issues with Classical Mechanism Design

3.1 Valuation Complexity

In any direct revelation mechanism, each agent's strategy is to report a type (truthfully or not). In complex settings like CAs and MAAs, an agent's type is an entire valuation function, i.e., a value for each of an exponential number of bundles of items or attribute configurations. In many cases determining these values will require complex and expensive computation on the part of the agent. This is not only true when the valuation function has an exponential size; even in standard one-item auctions computing the exact valuation for the item can be very costly. For example, for a telecom company to estimate the value of a license in particular area it must conduct a market study and develop a business plan, which not only incurs a cost but can also be inaccurate. However, in both cases, it is not always necessary to have complete and exact information about all agents' valuations to determine the efficient outcome of an auction. And when it is necessary, it may not be worthwhile: if the cost of further elicitation exceeds the increase in efficiency then one might settle for an approximately efficient outcome.

Using proxy agents is a common way to address the valuation problem. The goal is to have a proxy agent (one for each agent) bid on behalf of its owner when it has sufficient information to do so, and elicit more valuation information from the agent when it is both necessary and worth it. Parkes (2004) and Larson & Sandholm (2001a; 2001b; 2004; 2003) have studied two possible cost models for preference elicitation: free but limited (there is no cost attached to a single query but the total number of queries is limited) and unlimited but costly (each query has a fixed cost C but there is no global limit). Both consider various single item auctions where agents' true valuations are uniformly distributed between known bounds and queries are used to refine these bounds. The main difference between them is in their modeling of how an agent's computation refines the valuation information. Parkes uses a relatively general mathematical model where each query reduces the difference between the upper and lower bounds by a fixed ratio α such that the true valuation remains uniformly distributed between the new bounds. Larson & Sandholm consider a more informative model of valuation computation by assuming that each agent has a "performance profile" (one for itself and one for each of the other agents) describing precisely how computation will reduce valuation uncertainty. So, in their model, agents can not only refine their own valuation but also gain information about the others' in order to determine their bidding strategy.

Given these assumptions, both investigate how taking elicitation costs into account effects agents' strategies in various standard auctions, and then use this to analyze the loss in efficiency. Loss in efficiency is caused, of course, by the loss in utility to each agent due to elicitations costs, as well as a possible sub-optimal allocation of the item caused by the effects of costly elicitation on bidding strategies.

Parkes does not present a full game-theoretic equilibrium analysis but focuses on deriving good heuristics for elicitation strategies. Given these strategies, he compares the loss in efficiency across different auction formats and concludes that the Ascending-Price auction (a variant of the English auction) dominates both a sealed bid auction and a "posted-price" auction (a sequence of take it or leave it offers at a fixed price). Larson & Sandholm develop a full equilibrium analysis under costly elicitation, but their focus is entirely on what they call strategic computation, i.e., when an agent computes on other agents' valuation problem. They show that this does not occur (in equilibrium) in English and Vickrey auctions in the free but limited model (Larson and Sandholm (2001a)), but does in all standard auctions in the unlimited but costly model (Larson and Sandholm (2001b)). They also analyze the loss in efficiency of the Vickrey auction in the latter model (Larson and Sandholm (2003)) but again focusing only on the loss due to computation on others' valuations. They show how an auctioneer controlling computation costs (e.g., the auctioneer charges the agents for computing on its server) can design these costs to provide incentives for agents to choose computing strategies leading to optimal social welfare.

Both Parkes's and Larson & Sandholm's work show the need to take valuation complexity into account when designing mechanisms, even for simple one-item auctions. Parkes advocates the use of sequential, price-based mechanisms to reduce elicitation costs as well as focus the elicitation on relevant agents.

3.2 Communication Complexity

Assuming each agent somehow knows its own valuation function precisely and completely, the first step of the GVA is to report this valuation to the auctioneer. In one-item auctions, this means communicating a single real number. Although this might not seem excessive, in some settings, such as repeatedly auctioning low level resources like bandwidth or CPU time, the communication cost can be significant compared to the value of the resources. Full communication is also a problem with respect to privacy issues. Agents are generally reluctant to provide exact valuation information to their competitors. Furthermore, in a combinatorial or multi-attribute setting, full communication could potentially require a message of exponential size (one value for each bundle or each attribute configuration). A lot of the research in this area is concerned with designing other mechanisms, either sequential or direct with restrictions on bidding, that would require less communication while maintaining (sometimes approximate) efficiency.

Nisan and Segal (2003) show that this cannot be done in general. More precisely they show that, for a very general class of preferences[3], in any social choice problem with privately known preferences, finding a weakly Pareto efficient allocation is equivalent to discovering a set of competitive equilibrium prices supporting that allocation[4], which, in a combinatorial auctions requires, in the worst case, the communication of *at least* one price per bundle (and there are exponentially many bundles). They show that just achieving higher expected social welfare than that obtainable by simply auctioning all goods in a unique bundle requires exponential communication when the number of goods is substantially larger than the number of agents. Of course it should be noted that these are worst case results; depending on the actual preferences of agents the problem can be much simpler, as we will see several times in this document, particularly in Section 5.2.2.

Naturally, communication complexity is not unrelated to the valuation problem described above. Restrictions on communication can simplify the valuation problem since agents only need to compute values that they are allowed to communicate. Conversely, some forms of restriction on valuation will restrict communication since an agent cannot communicate information it does not have.

3.3 Computation Complexity

Once agents' bids have been submitted, either by direct revelation like in the GVA or through any sequential mechanism, the auctioneer must determine the final outcome of the auction (which agents win and what the payments are) so as to maximize his objective (e.g., social welfare, seller's revenue).

[3] Their result applies to valuations that are bounded, satisfy free disposal, and only depends on the agent's own allocation and not on any externalities.

[4] Competitive (or Lindahl) equilibrium prices are a set of prices for each agent and for each bundle. A Lindahl equilibrium supports an allocation if the allocation maximizes the agents' (seller included) utility at these prices.

In some settings this will be very easy: in a single-item auction the winner is the agent with the highest reported valuation. In a combinatorial auction, however, it is much more complicated: finding the welfare maximizing allocation of goods to agents (the second step of the GVA) is known to be NP-complete (Rothkopf *et al.* (1998)), by reduction to a set-packing problem. In fact, as we will see, even finding a "good" approximation cannot be done in polynomial time, by reduction of Max-Clique which is inapproximable (Hastad (1999)).[5]

Three different approaches have been adopted to deal with computation complexity: designing good heuristics so that (optimal) winner determination will be feasible in most *practical* cases; identifying types of structure in the problem that make it feasible (Sections 4.1.1 and 4.1.3); and using tractable approximation techniques that are efficient in practice (Section 4.1.2). The latter approach is complicated by the fact that using an approximate allocation in the GVA is often not compatible with truthful implementation.

3.4 Prior Independence

A large part of mechanism design is concerned with designing mechanisms that implement a social choice function no matter which agents are going to participate, that is, without taking into account any prior knowledge the designer might have about the agents' types. It seems natural that, when extra information is available, using it would enable the designer to further optimize his objective. This is in fact the approach taken by Myerson (1981) for the revenue-optimal auction described later. Instead of a mechanism per se, Myerson proposes an algorithm for constructing a mechanism tailored to a specific, commonly known prior over types in order to maximize the seller's revenue. Using prior information can also allow us to sidestep Impossibility results such as Gibbard-Satterthwaite and Myerson-Satterthwaite by waiving the condition that the mechanism must be applicable for all possible types, since we are restricting ourselves to types consistent with the priors. In the computer science literature, this approach of using (probabilistic or non-deterministic) prior information is called Automated mechanism design and is the subject of section 6.

4 Improving VCG Mechanisms

4.1 Computational Improvements

As we have seen in Section 3.3, given a set of bundle bids from each bidder, determining which bids should win in a combinatorial auction is NP-complete. Since the GVA requires that this be done $n + 1$ times (for n bidders), it is critical to address this issue at least in practical settings. As mentioned, there are

[5] See section 4.1.2.

two main approaches to reducing the computational complexity of the GVA: designing algorithms which, despite their exponential worst-case behaviour, in practice find optimal allocations quickly; and trying to find good approximations in polynomial time. It turns out that not only is the latter also impossible in the worst-case, but approximating winner determination is not easily combined with incentive compatibility.

4.1.1 "Faster" Optimal Allocation

Initial attempts to use search techniques specifically designed for winner determination in CAs (Sandholm (1999); Fujishima *et al.* (1999)) were not empirically successful since it was discovered (Andersson *et al.* (2000)) that simply formulating the problem as an integer program (IP) and solving it with commercial software such as CPLEX gave better results. A more sophisticated search algorithm was proposed in Sandholm *et al.* (2001). CABOB tries to find the optimal allocation by branching on bids in a Depth-First search with Branch and Bound. The algorithm maintains a "bid graph", i.e., a graph where the nodes represent the submitted bids that have not been branched on and an edge between two bids indicates that the bids have at least one item in common. The bid graph is updated throughout the search, removing bids and corresponding edges when they are temporarily assigned to the optimal solution, and replacing them after backtracking. Upper and lower bounds, used for Branch and Bound throughout the search, are computed by relaxing the IP formulation into an Linear Program (LP), which is much easier to solve. The paper presents a number of bid ordering heuristics for the depth-first search, based, among others, on the number of items in the bid, or on the average price per item. Empirical results suggested dynamically selecting one of these heuristics according to the density of the LP relaxation.

There is no obvious way to evaluate winner determination algorithms. Since they all have exponential worst case complexity, an algorithm will be deemed better than the other if it is faster when used on common instances of the bidders' valuation functions. However, which valuations are common in a CA setting is a matter of conjecture since CAs are only recently being used in practice. A test suite of distributions for combinatorial valuations, CATS, has been developed (Leyton-Brown *et al.* (2000)), containing, if not common, at least "common sense" distributions. Sandholm *et al.* (2001) compares CABOB to CPLEX 7.0 on CATS and other distributions. Empirically CABOB is often slightly better than CPLEX, never much worse and sometimes much better. Results also show good anytime performance which suggests that CABOB could be used to approximate the winner determination problem. However, as we will see in the next section, approximating the allocation in a CA creates many other issues.

4.1.2 Approximation Techniques

As mentioned above, finding a good approximation[6] cannot be done in polynomial time in all cases. Nevertheless there has been a lot of research into how polynomial approximations might be used in CAs. The main issue is that, as we will see, the incentive compatibility guarantee of VCG mechanisms fails when using a non-optimal allocation function.

Lehman *et al.* (2002) consider a very restricted setting where all agents are single-minded, that is they only value one particular bundle each. Even in this case, optimal allocation is NP-complete. They propose a very simple greedy allocation algorithm and show that the VCG mechanism based on this allocation is not incentive compatible for single-minded agents. Given this negative result, their approach is to move away from VCG-based mechanisms and to try to find other types of mechanisms that do satisfy IC. In fact they present four simple and intuitive conditions on a mechanism for single-minded agents that imply IC and propose a payment scheme which, along with the greedy allocation algorithm, satisfies them. They show however that if agents are not single-minded, no payment scheme will make their greedy allocation incentive compatible.

Nisan and Ronen (2000) show a much stronger negative result: any approximation algorithm which, if a particular item is valued by only one agent, always allocates this item to that agent, will make the corresponding VCG mechanism non-incentive compatible. In other words, any VCG-based [7] incentive compatible mechanism would provide a very poor approximation of the optimal allocation. Another approach is to somewhat relax the requirement of incentive compatibility to one that is still reasonable and is not incompatible with good approximation algorithms. A very interesting property of VCG-based mechanisms that will prove to be very useful here is the "self-correcting" property: in a VCG-based mechanism, given a report of the other agents, if an agent has a possible false report (a lie) that would yield higher utility than a truthful report, then this lie will also produce an allocation with higher social welfare. This is because finding a "good lie" comes down to solving the whole allocation problem. The approach taken in Sanghvi and Parkes (2004) is to settle for a mechanism in which it is computationally hard for the agents to lie. More formally, they require that finding a report that increases an agent's utility by at least ϵ compared to reporting the truth, given the types reported by the agents, is NP-hard. They prove that this is always possible: given any approximation algorithm for the allocation problem, it can be transformed in linear time and with no loss in efficiency, into one that satisfies certain basic properties (such as not leaving goods unassigned, not assigning them to agents who do not want them, and providing an allocation that does no worse than simply giving any single agent his most preferred bundle), and these properties imply that manipulation of the corresponding VCG-based mechanism is NP-hard.

[6] Formally, with n agents, an approximation ratio smaller than $n^{1-\varepsilon}$ cannot be achieved in polynomial time.

[7] A mechanism is called VCG-based if its payment rule is that of the VCG mechanism but its allocation rule is different.

Note that this result only states that lying is difficult *in the worst case*: it is very possible that in many instances there is a polynomial time algorithm that can find a good lie. In those cases, if the agents use the same approximate algorithm as the mechanism then the mechanism will find the optimal allocation and truth-telling will be a dominant strategy. Otherwise, it could be the case that the mechanism's algorithm does not find the optimal allocation but some of the agents have a different polynomial algorithm that is more efficient on this particular instance and can therefore provide them with an ϵ-better lie. In this latter case, it would be useful to modify the mechanism so that it could take advantage of this extra knowledge some of the agents have.

Although it is not presented quite this way, this is what Nisan and Ronen (2000) do. Their approach is also to relax the concept of incentive compatibility but to one stronger than Parkes's worst-case approach. They introduce the idea of a *feasibly-dominant* action, that is an action that is a best response to any actions of the others, *as far as the agent knows*. In other words, there could be a better response to the others but the agent cannot compute it. Each agent's knowledge is represented in an "appeal function", which is basically the agent's current estimate of his best-response function, i.e., a mapping from types reported by the others to the type it should report to maximize his own utility. But since finding a good report for an agent comes down to finding a good global allocation, this appeal function can be thought of as the agent's own approximate allocation algorithm. Nisan and Ronen's Second Chance mechanism tries to combine the power of all these algorithms to increase efficiency while maintaining some version of Incentive Compatibility. The Second Chance mechanism allows agents to report both their type and their appeal function. The mechanism then computes the (approximately optimal) allocation based on the reported types and again once for each appeal of the agents, and picks the one with the highest social welfare (based on reported types). Payments are then computed as in a VCG mechanism. In other words, each agent reports something like "Here's my true type but if the others report this, then I really think your algorithm would do better if I lied this way" and the mechanism checks whether or not the agents are right. This mechanism also satisfies individual rationality (i.e., participation constraints). They prove that reporting one's true type and appeal function is a feasibly dominant strategy in the Second Chance mechanism. Compared to a VCG-based mechanism using the same allocation, the Second Chance mechanism satisfies some reasonable notion of incentive compatibility and is able to combine the results of several approximation algorithms to pick the best one. Of course, depending on how compactly the appeal function can be represented, the Second Chance mechanism can significantly increase the (already substantial) communication requirements.

4.1.3 Special Cases

Since the complexity results on winner determination are all concerned with worst-case instances, there has been significant research into identifying ranges of problems where winner determination is manageable, i.e., when there is a polynomial time algorithm that solves it optimally. The idea is that some types of structure imposed on or exhibited by the set of submitted bids can make allocation easy. If the agents' valuations happen to conform to this structure, which in some specific domains will not be an unrealistic assumption, then the VCG mechanism can be used. However, if this structure is imposed on the bids but is not exhibited by the true valuation functions of the agents then, even if one could define truthful revelation (e.g., as revealing the true valuation's projection on the allowed bid space), it would not, in general, be a dominant strategy. Rothkopf *et al.* (1998); Nisan (2000); Tennenholtz (2000) all present various types of structure on bids that make winner determination manageable. None of these papers, however, deal with the incentive issues that arise when such structure is imposed, a subject that we address further in Section 4.2.2.

4.2 Communication Improvements

There are two approaches to reducing the communication complexity of direct revelation mechanisms such as the GVA. The first is to maintain full revelation of the valuation function, but design representations for these valuations that allow for a more compact encoding, at least for common valuation functions. The second approach is to restrict revelation to incomplete but compact languages (which can also help reduce valuation complexity) while minimizing the impact of those restrictions on bidding complexity and loss in efficiency.

4.2.1 Complete Languages for CAs

In the CA setting with m goods, a valuation function is a mapping from 2^m to $\mathbb{R}$. Even with finite precision for real numbers, such that $|\mathbb{R}|$ is finite, the number of possible valuation functions would be $|\mathbb{R}|^{2^m}$. It is therefore impossible for one bidding language to represent all (even most) valuations compactly. Instead what we require of a complete bidding language is that it can represent all possible valuations (full expressiveness), that "important" valuations are represented compactly, and that the language is easily interpreted. Unfortunately there is no formal definition of an "important" valuation, it will generally depend on the context. There are however several examples of valuation structures that are used to compare different bidding languages (Nisan (2000)).

For example, a *symmetric valuation* is one where the bidder does not differentiate between goods so the value of a bundle only depends on the number of goods. The *general symmetric valuation* associates values $p_1, p_2, \ldots, p_m$ to the first, second,..., m^{th} item won, respectively, thus for any bundle S, $v(S) = \sum_1^{|S|} p_j$. If $p_j = 1, \forall j \leq K$ and 0 otherwise, this is the *K-budget valuation*; if $p_{m/2} = 1$ and 0 otherwise, it is the *majority valuation*. In general, single items could have different values. An additive valuation is one where the value of a bundle is the sum of the values of the items in the bundle.

A very commonly made assumption about combinatorial valuations is that of *free disposal*, that is that the valuation of a subset of items does not decrease when an additional item is added. A language is often still considered fully expressive if it only represents valuations with free disposal. Combinatorial bidding languages can be classified into three types (following Boutilier and Hoos (2001)) based on whether or not they allow logical combinations of goods as bids, $Ł_G$, logical combinations of bundle bids, $Ł_B$, or both (i.e., Generalized logical Bids), $Ł_{GB}$. In $Ł_G$ languages (Hoos and Boutilier (2000)), a bid is a pair (α, p) where α is a logical formula with items as atoms and p is a price. A valuation will often be represented using several logical bids, with independent preferences represented in separate bids. $Ł_G$ languages are fully expressive (with free disposal) but are particularly useful because of their compact representation of disjunction. For example if there two types of items, g and h, and two items of each type, a user might want at least one item of each type which can be expressed as $\langle (g_1 \vee g_2) \wedge (h_1 \vee h_2), p \rangle$ in $Ł_G$.

In $Ł_B$ languages, an *atomic bid* is a pair (S, p) where S is a subset of items and p is a price, and an $Ł_B$ bid is a logical combination of atomic bids. The OR language accepts a collection of atomic bids and implicitly assumes that the bidder values any union of disjoint bids as the sum of their prices. The OR language is not fully expressive since it cannot represent any substitutability among items. The XOR language implicitly assumes that the bidder will accept at most one of the atomic bids submitted. The value of any subset is that of the most valued atomic bid in that subset. XOR is fully expressive (with the free disposal assumption) and easily interpretable but even some simple valuations such as the symmetric additive valuation require bids with size exponential in the number of items (as opposed to linear for OR). Many languages involving combinations or ORs and XORs have been proposed, but all are dominated by the following language. Nisan (2000) introduced the OR* bidding language, which is identical to OR except that bidders may add "dummy" (worthless) items that enable the representation of substitutability. OR* is fully expressive and at least as compact as any combination of ORs and XORs. Unfortunately there are still some common valuations that are not compact in OR*, such as the majority valuation. Although it dominates any other $Ł_B$ language in compactness, OR* is not easily interpretable: answering value or demand queries is NP-complete because a subset must be optimally allocated to the atomic bids to determine its value. [8]

The last language type, $Ł_{GB}$ (Boutilier and Hoos (2001)), is a combination of both $Ł_G$ and $Ł_B$ languages. Bids in $Ł_{GB}$, called Generalized Logical Bids (GLBs), are logical combinations of bids *and* goods, meaning that prices can be associated with any subformula. Any of the $\vee$,$\wedge$ and $\oplus$ connectives can be used to combine good-price pairs. $Ł_{GB}$ is fully expressive (with free disposal) and is more compact than any $Ł_G$ or $Ł_B$ language: any bid expressed in OR, OR*, XOR or $Ł_G$ in size s can be expressed in size at most s in $Ł_{GB}$. The converse, however, is not true: some valuation functions are represented exponentially more compactly in $Ł_{GB}$ than in any other language. For example, in a scenario with

[8] This is true of any language using ORs, but not of XOR.

both sharable resources (a machine m) and consumable resources (raw material to be processed by the machine, $r_1, \ldots, r_k$) where the machine and the material are worthless alone, and different types of material have different value, the valuation can be concisely expressed as: $\langle m \wedge r_1, p_1 \rangle \vee \langle m \wedge r_2, p_2 \rangle \vee \ldots \vee \langle m \wedge r_k, p_k \rangle$. In any $\mathrm{Ł}_B$ language, this would require a bid for each subset of consumable goods (e.g., $r_1; r_2; r_1 r_2; \ldots$) of which there are exponentially many. $\mathrm{Ł}_{GB}$ is concise in general because most complex preferences have a lot of structure and $\mathrm{Ł}_{GB}$ directly exploits this structure. For this reason, $\mathrm{Ł}_{GB}$ also has low "expression complexity", that is it is often natural for a user to write down her preferences in this language. $\mathrm{Ł}_{GB}$ is also easily interpretable (answering value queries is straightforward). Boutilier (2002b) proposes an IP formulation for the winner determination problem with input bids specified in $\mathrm{Ł}_{GB}$ that does not involve converting these bids to "flat" (XOR) bids. Empirical results show that this is not only much faster than the standard winner determination IP for flat bids, but it can solve much larger problems. Although there is no direct comparison, the structured IP seems to dominate the standard IP more than CABOB (Sandholm *et al.* (2001)) did.

4.2.2 Restricted Languages

Since no complete bidding language will represent every valuation compactly, one might try to enforce compactness by restricting the bids to incomplete languages, i.e., languages that cannot represent all valuations, compactly or not. Note that the special structures mentioned in Section 4.1.3 are all incomplete languages, but the papers surveyed in that section were only concerned with computational aspects of winner determination and ignored both strategic considerations and communications requirements.

Ronen (2001) studies the impact on incentive compatibility of using restrictions such as those presented in Section 4.1.3. With such incomplete languages, the standard VCG mechanism, even with an optimal allocation function, can be manipulated, i.e., truth-telling is not necessarily dominant. Ronen attempts to "fix" this the same way Nisan and Ronen "fixed" VCG-based mechanisms with approximate allocation algorithms with the Second Chance mechanism. In this setting, each agent submits a description of its valuation function in the incomplete language imposed by the mechanism, as well as an oracle which can answer any query about the the agent's true valuation, and an appeal function describing how the agent would act if it knew the others' reports before reporting itself. The mechanism then finds the optimal allocation with respect to the incomplete descriptions submitted by the agents, and then again for each appealed description. It then picks the allocation with greatest social welfare with respect to the valuations represented by the oracles. Payments are then computed using the VCG formula.

As in Nisan and Ronen's work, if the appeal function has a sufficiently simple form, i.e., a mapping from other agents' reports to an agent's own oracle and description report, then revealing one's true description, oracle and appeal function is feasibly dominant. Ronen (2001) also mentions the use of a consistency checker, an algorithm that checks in a polynomial number of queries to the oracle whether the description is consistent with the oracle. It is not clear when this consistency checker is used by the mechanism nor whether it is necessary for truth-telling to be feasibly dominant.

The main advantage of this mechanism is that the incomplete language can be chosen so that winner determination can be done in polynomial time, thus addressing the computational problem of the VCG mechanism. In terms of communication and valuation complexity, things are not so clear: valuation descriptions are compact because the language was chosen that way, but submitting an oracle is equivalent to full revelation and therefore unreasonable, as in the standard VCG. This mechanism, however, is such that the oracle is only queried a polynomial number of times. It is not made clear in Ronen (2001) whether or not addressing these queries to the agent directly (without submitting an oracle) would maintain truth revelation as a feasibly dominant strategy. In the affirmative, it seems that such modifications would greatly improve the usefulness of their mechanism.

Restrictive bidding languages are naturally not only useful to address computational concerns, but also, and mostly, to reduce communication and revelation requirements. This is the approach of Holzman et al (2003): they consider bidding languages that only allow bids on some (incomplete) family of bundles Σ. As previously mentioned, such restrictions will in general be incompatible with incentive compatibility so they only require that truth-telling be an ex-post equilibrium. More formally, for any valuation function, they define its projection on the space of Σ-valuations, i.e., valuations where the value of any bundle is that of the highest valued bundle in Σ it contains. A bundling equilibrium is an ex-post individually rational equilibrium where each agent's strategy, f^{Σ}, is to report the Σ-projection of its true valuation function. They prove that f^{Σ} is a bundling equilibrium if and only if Σ is a quasi-field. In particular, if Σ is the field generated by a partition of the items then f^{Σ} is called a partition-based equilibrium.

There are naturally several such bundling equilibria, each generating an allocation with possibly different social welfare values, and the question is how to pick Σ so as to maximize the Social Welfare. Of course this value will in general not be the optimal social welfare attained by the truth-telling dominant strategy equilibrium. They provide upper bounds on the worst case loss in social welfare incurred by restricting bidding to Σ. Except for partitions of very small size, their bounds are not very useful: the upper bound on welfare loss increases as the communication restrictions decrease! In some cases, they prove that these bounds are tight, thus illustrating that it is possible to pick a partition with a very bad worst-case loss.

The idea of Holzman *et al.* (2003) is to reduce communication while maintaining some form of incentives for truth-telling. In the introduction, they claim that a designer only needs to *suggest*, as opposed to *impose*, the partition Σ for the corresponding bundling equilibrium to be adopted. It seems, however, questionable whether a rational agent would pick an ex-post equilibrium strategy over a dominant one unless forced to or unless those strategies are identical (in which case the true valuation is its own projection and any communication gain would have been obtained in a classic VCG mechanism). So it is only realistic to consider bundling equilibria when bidding restrictions are imposed on agents, in which case it would be interesting to know whether these equilibria become dominant strategies in the game with restricted actions.

5 Sequential Mechanisms

The previous section focused on direct revelation mechanisms, in particular VCG mechanisms or variations thereof, that attempt to address some of the computational and communication issues. Full revelation is not always necessary to guarantee optimal efficiency. For example, to assign a single item efficiently, the identity of the agent with the highest valuation is sufficient information, regardless of what that value is. However neither the designer nor the agents know exactly what information is relevant before at least some information is revealed. This suggests moving away from one-shot direct revelation mechanisms and towards sequential elicitation through queries in order to use previously revealed information to better decide what to elicit next. We first look at price-based mechanisms, i.e., mechanisms that only use price (demand) queries, before studying more general types of centralized elicitation.

5.1 Price-based mechanisms

5.1.1 One Item Auctions

Although the communication requirements of a one-item auction such as Vickrey or Myerson might seem very low — a single real number must be sent by each agent — there are settings, such as auctions for low-level resources, where the value of the items auctioned is comparable to the cost of sending a real number, and therefore limited precision may be desirable. Using limited-precision bids not only reduces communication but the valuation complexity and the information revelation of the agents as well. Blumrosen and Nisan (2002) study limited precision one-item auctions in simultaneous mechanisms, before extending them to sequential mechanisms (Blumrosen *et al.* (2003)). They impose a hard limit of only k possible messages that the agents can choose from and aim to find such a limited-precision mechanism with a dominant strategy equilibrium that minimizes the loss in expected social welfare or revenue compared to a full-precision auction. They introduce priority games (PGs), a class of one-item auction mechanisms, where agents choose one message out of a set of k ordered messages and the item is allocated as follows: if there is an agent whose bid (message) is strictly greater than all the others, then this agent wins, otherwise the tie is broken according to a pre-specified order on the agents (e.g., Agent 1 has priority over Agent 2). Payments are such that any losing agent pays zero (thus ensuring individual rationality) and the winner pays the minimum valuation it could have had and still win the item by bidding truthfully. They prove that priority games have a dominant strategy equilibrium where each agent's strategy is a threshold strategy: there is a set of real numbers partitioning the valuation space such that an agent's optimal bid is to report the highest threshold that lies below its true valuation (i.e., report j such that $x_j \leq v_i < x_{j+1}$). The payment rule of a PG induces a unique dominant threshold strategy for each agent. So a PG is a limited precision auction where each agent truthfully reports a lower bound on its valuation, picked from a set of fixed possible reports.

Clearly, limiting the precision will, in expectation, incur a loss in social welfare, compared to the Vickrey auction because of the tie breaking rule which will not always allocate the item to the agent with the highest valuation. The next question they address is how to pick the thresholds of a PG so as to minimize this expected loss in welfare. They optimize these thresholds using a prior distribution on agents' valuations and show that in the case of 2 players, mutually-centered thresholds (where an agent's threshold for bid j is set to the expected valuation of the other agent *given* that he also bids j) are optimal. They also show that a priority game with mutually-centered thresholds minimizes the expected welfare loss among all mechanisms restricted to k messages per agent (even those without a dominant strategy equilibrium) and that this loss is $O(1/k^2)$. Finally, they show that with thresholds that divide the valuation space into equal partitions (and therefore independent of the prior) the expected loss given any prior is at most $1/k$.

When the objective is seller's revenue, they transpose all of these results by noting that revenue maximization is equivalent to virtual utility maximization, as in the Myerson auction,[9] and modify the definitions of priority games and mutually-centered thresholds accordingly.

In Blumrosen *et al.* (2003), they extend this approach to sequential mechanisms for limited precision one-item auctions. In their model, a sequential mechanism is one where agents send several smaller messages (i.e., containing fewer bits) each in a pre-specified order and where all bits that have already been sent are publicly known. They give an example of a sequential mechanism that attains higher expected social welfare than the optimal priority game, although this is done in Bayes-Nash equilibrium and not in dominant strategies. They show, however, that the communication gain that can be achieved by sequential mechanisms is at best linear in the number of agents: with n players, a simultaneous mechanism can achieve the same expected social welfare as any sequential mechanism that uses m bits, with "only" about $n \cdot m$ bits used.[10]. They seem to view this as a negative result that does not justify investigating sequential mechanisms further.

Kress and Boutilier (2004), on the other hand, study sequential mechanisms in more details by considering a restricted class of mechanisms and showing how to optimize them to minimize the welfare loss. They define incremental mechanisms as sequential mechanisms where the space of possible valuations of an agent is refined at every iteration. They consider incremental mechanisms where, as in Blumrosen

[9] See Section 6.

[10] $n \cdot m - n(n-3)/2$ to be precise.

and Nisan's work, at each round the agents have only a finite set of ordered messages to choose from. Unlike Blumrosen and Nisan, they require that an allocation be made only when the mechanism is certain that the winner has the highest valuation. They prove that, under such restrictions, any deterministic mechanism with a dominant strategy equilibrium and ex-post individual rationality must be an increasing price mechanism, i.e., a mechanism where, for fixed strategies of the others, the price paid when adopting a strategy that wins at the earliest possible round is no more than the price paid using a strategy that wins at a later round.

Such an incremental mechanism, the "bisection auction", has been proposed in the economics community (Grigorieva *et al.* (2002)) but with a different motivation: they assume that all agents' exact valuations are in some finite set (e.g., the integers within some bounds) so their goal is not implement a limited-precision auction but to remedy the slowness of a first-price auction and the excessive information revelation and valuation complexity of a Vickrey auction. The "bisection auction" is basically a binary search in this finite-precision valuation space for both the highest and second highest valuation, in a way that maintains truthful revelation as a dominant strategy. Note however that if used as a limited precision auction, i.e., when agents valuations can lie outside the set of finite precision set, the bisection auction loses its dominant strategy property (for the same reasons a limited precision Vickrey auction with random tie-breaking does).

The main auction mechanism proposed in Kress and Boutilier (2004) is ASIA, for Adaptive Symmetric Iterative Auction. ASIA works as follows: at each round the mechanism posts a price, with the constraint that this price strictly increases, and players can choose to drop out (send 0) or stay active (send 1). The auction terminates when either only one player is active, in which case he wins and pays the last posted price, or none are, in which case no allocation is made.[11] They show that it is dominant for agents to play a threshold strategy based on their true valuation, which in this context means bidding one if the posted price is lower than one's valuation and zero otherwise.

In ASIA, all allocations of the good are optimal (i.e., to the agent with the highest valuation), but there are cases when the good is not allocated at all. This happens if the posted price "jumps" above the highest valuation, which suggests that prices should not be increased too fast. A small price increment, on the other hand, means higher communication because less agents will drop out of the auction early. They model this sequential decision problem (the pricing rule) as a Markov Decision Process (MDP) in order to use the prior on valuations to find the optimal trade-off. They compare the optimal policy given by the MDP to a simple pricing heuristic. Not surprisingly the MDP policy performs much better. Their experiments also show how increasing the communication costs makes the average number of bits sent per player decrease, but the average welfare loss increase. They also show how the same level of welfare loss can be achieved with fewer bits when the number of players increases.

[11] This is basically a Japanese auction.

They then consider a stochastic version, STASIA, that reduces the social welfare loss in ASIA by making sure an allocation is always made. The threshold strategy with one's true valuation is still dominant. Their experiments compare ASIA and STASIA under several pricing rules and show that STASIA incurs a smaller social welfare loss (because the object is always allocated but not necessarily to the agent with the highest valuation) with roughly the same communication costs.

5.1.2 Combinatorial and Multi-Attribute Auctions

Designing an iterative version of the VCG mechanisms for CAs has received significant attention in the economics literature, but the proposed solutions have so far only addressed special cases with restrictions on valuation functions, like unit-demand or gross substitutes (Kelso and Crawford (1982); Demange *et al.* (1986b); Gul and Stachetti (2000); Ausubel (2000)). Wurman and Wellman (2000)'s AkBA family of auctions apply to unrestricted preferences, but does not provide any truth incentives. Parkes and Ungar's work addresses the general problem of finding an efficient iterative auction (Parkes and Ungar (2000a)) with an incentive for truthful revelation (Parkes and Ungar (2000b); Parkes (2001)). Their approach is based on the strong relationship between combinatorial auctions and primal-dual optimization theory explored by Bikhchandani and Ostroy (2002).

Parkes and Ungar (2000a) introduce iBundle, an efficient iterative combinatorial auction which works as follows: at each round, agents can submit XOR bids on bundles of items, which the auctioneer uses to compute (and announce) a provisional allocation. The auctioneer maintains a set of ask prices (one for each bundle that has received a bid). At the end of a round, a bundle's price is increased if at least one agent not in the provisional allocation has bid above the previous ask price and the new ask price is set to ϵ (the minimal bid increment) above the highest such losing bid. The auction terminates when either all agents have repeated their previous bids, or when all agents are assigned a bundle in the provisional allocation. They prove that iBundle terminates with an efficient allocation as the bid increment tends towards zero, when agents play a truthful myopic best-response strategy. A (truthful) myopic best-response strategy is to bid, at each round, for the bundles that maximize one's (true) utility *at the current ask prices*. They also prove that such strategies form an ex-post equilibrium and not a dominant strategy equilibrium, leaving the auction at least theoretically open to manipulation. Computationally, iBundle must solve a Winner Determination problem at each round, as opposed to only the last round in the VCG auction, but these instances will be much smaller than in VCG since there are much fewer bids submitted at each round. Experimental results show that iBundle is faster than the VCG except on very small problems. Increasing the bid increment will reduce the number of rounds before iBundle terminates but at some cost in efficiency. Interestingly, using approximate allocation algorithms, at any round but the last one, does not change iBundle's incentives properties, unlike in VCG, since the auction must terminate with an efficient allocation. Communication complexity is similar: agents will send fewer bits per round but there can be many rounds; again this will depend on the problem and the communication gain will, in general, grow with the size of the problem. Ask prices for bundles are only posted when they are increased, and, with n agents, only $n-1$ prices can increase at each round due to the update rule.

Parkes and Ungar (2000b); Parkes (2001) attempt to address manipulation in iBundle. The first step is to add proxy agents, one for each bidder, who use a myopic best-response strategy based on queried valuation information. The goal is to make truthful revelation to the proxy agents a dominant strategy. They prove that this would be the case if the auction implemented a Vickrey outcome: an efficient allocation, which iBundle already provides, and Vickrey payments, which it does not in general. Note that they make a distinction between revelation that takes place before the auction starts, and dynamic revelation where queries could be asked during the auction. The dominant strategy property above holds for the former, while the latter case only supports an ex-post equilibrium, and is therefore theoretically subject to manipulation, albeit computationally hard since it would require exploiting past posted prices to infer information about other agents' preferences and use that information to improve one's utility with responses to queries consistent with previously revealed information. The goal now is to modify the proxied iBundle auction so that it computes Vickrey payments.

Parkes and Ungar (2000b) proposes to add a price adjustment step, Adjust*, following iBundle. Adjust* independently adjusts each agent's iBundle payment towards Vickrey payments by giving agents their Vickrey discount: the increase in social welfare caused by that agent's presence. The adjusted price is not always equal to Vickrey payments because iBundle's prices are not necessarily the agent's valuation. They prove however that in a restricted case Adjust* does compute Vickrey payments and therefore that the proxied iBundle+Adjust* auction has (non-dynamic) truth revelation as a dominant strategy equilibrium. The sufficient condition for this to hold is that, when there is more than one agent in the optimal allocation, if one of those agents is not in one of the second-best allocations then he must have bid his true valuation for the bundle he received. What they call the second-best allocation is actually the best allocation in the sub-problem with that agent removed, which must be computed for each agent by Adjust*, thus adding more computational complexity. Unfortunately this sufficient condition will not hold in general when iBundle terminates.

Parkes (2001) proposes another extension of iBundle that attempts to handle the cases when the condition does not hold. This new auction first runs the proxied iBundle+Adjust* auction and checks whether there are agents in the optimal allocation that are not in all the second-best allocations. If there are none then the condition above must hold and the auction terminates (with the Vickrey outcome). Otherwise, they introduce dummy agents and continue to run iBundle in order to elicit more precise information and adjust prices further towards Vickrey payments. They conjecture that this auction always implements the Vickrey outcome, although they have not yet proved it, and give experimental results supporting their claim. If confirmed, this would an efficient iterative auction where the only possible manipulation would be by adapting one's valuation revelation to the proxy agents based on the (very small amount of) information available about other agents' bids. The auction also permits the use of approximate algorithms (at the cost of increasing the number of rounds) thus allowing a tradeoff between computational and communication complexity.

The ideas presented above for combinatorial auctions have also been partially transposed to the design of iterative mechanisms for the multi-attribute (reverse) auction problem in Parkes and Kalagnanam (2004); Sunderam and Parkes (2003).

5.2 Query-based mechanisms

In this subsection, we consider methods for centralized preference elicitation with more general types of queries. There are three main approaches: the first is to use heuristics that take advantage of structural properties of the problem to ask the most relevant queries; the second draws a parallel between preference elicitation and learning theory and applies results from the latter to the former; the third attempts to apply decision-theoretic techniques to elicitation.

5.2.1 Heuristic Search

Conen and Sandholm have explored the use of heuristic search techniques for centralized preference elicitation extensively in a series of papers (Conen and Sandholm (2001, 2002a,b)). The main idea is to take advantage of some inherent structural properties of CAs. They use two types of topological structure. A *rank lattice* represents a partial order on global preferences over allocations, i.e., the top node consists of each agent's most preferred bundle, nodes at the second level contain the second preferred bundle of one agent and the most preferred bundles of all others, etc... Of course many of these nodes will not be valid allocations because some bundles will have items in common. The non-dominated valid nodes left in the lattice are Pareto-efficient allocations. Elicitation queries are selected so as to prune invalid allocations and identify the welfare maximizing Pareto-efficient one. The second type of topological structure they use is the *augmented-order graph* where nodes represent agent-bundle pairs and edges reflect dominance of one bundle over another for that agent. The graph is augmented with rank information as well as bounds on value for each bundle. In both cases, they present procedures for efficiently querying agents and optimally propagating this information through the lattice or graph in order to reduce the number of candidate (potentially efficient) allocations (Conen and Sandholm (2001)). Their algorithms use combinations of value, order and rank queries (see Section 2.1.1) in the elicitation process. Note, however, that the use of rank queries, though very efficient in reducing communication, is debatable when valuation complexity is an issue. In some contexts, asking for an agent's most preferred bundle might be reasonable but asking for the i^{th}-most preferred one will generally not be an easy query to answer without computing the entire valuation function. The question of incentive compatibility has so far been ignored. This can be achieved by applying Vickrey payments at the end of the elicitation phase. However, computing these payments requires finding other efficient allocations and therefore might necessitate further elicitation. It turns out (Conen and Sandholm (2002b)) that, with some modifications, several of their algorithms obtain Vickrey payments for free; that is, all the information necessary to compute them has already been elicited while searching for the optimal allocation.

As previously mentioned, Nisan and Segal (2003) showed that the worst case communication complexity in CAs is exponential. Naturally heuristic search techniques cannot avoid such worst cases, but they might require less revelation on practical problems. Hudson and Sandholm (2004) tested several of the centralized elicitors described above on the CATS test suites (Leyton-Brown *et al.* (2000)) and the results are very encouraging in terms of communication complexity: in practice only a small fraction of the full valuation revelation is elicited, and this fraction decreases with the number of participating agents. Note, however, that these algorithms involve maintaining and propagating information through large data structures (rank lattice, augmented-order graph) making their computational complexity exponential.

5.2.2 Learning Theory

The second approach to centralized elicitation is based on the principles of exact learning from the query learning literature, where the goal is exactly to learn an unknown target function from a known class through queries to an oracle. Throughout the querying process, the algorithm maintains a manifest function which should eventually equal the target function. A class of functions is said to be *poly-query exactly learnable* if this can be done in a number of queries polynomial in the size of the target function and of its input space. It is said to be *efficiently exactly learnable* if the process takes *time* polynomial in the same variables. The simplest type of queries in learning theory are membership queries, which are exactly equivalent to value queries in preference elicitation. Much work has been done in the learning community to identify which classes of functions can be learned with membership queries, and some of these results have been transposed to valuation functions that can be elicited with value queries in Santi *et al.* (2004); Zinkevich *et al.* (2003); Blum *et al.* (2004). These classes include read-once formulae (basically functions represented as trees with nodes such as SUM, MAX or ALL), toolbox DNF (monotone polynomials) and k-wise dependent valuations (where interactions between items are restricted to sets of size at most k).

All of these papers focus on polynomial algorithms that learn a user's preferences completely. However, the real goal of preference elicitation and mechanism design is to make a decision on behalf of a user or a group of agents and the preferences of the agents are only relevant insofar as they influence this decision. The function that should be learned is therefore the decision function, i.e., the allocation and payment rules in a CA. This is the approach of Lahaie and Parkes (2004). Lahaie and Parkes characterize a class of valuation functions as poly-query (respectively, efficiently) elicitable if there is an algorithm that outputs an optimal allocation with a polynomial number of queries (respectively, in polynomial time). They introduce demand queries, where the elicitor provides a bundle of items and a set of prices – one for each bundle – to an agent, and the agent must either confirm that this is indeed his most preferred bundle given those prices, or, if not, provide a counter-example. They prove that demand queries with Lindahl prices (i.e., prices that support a competitive equilibrium given the current estimation of the valuation functions) correspond to equivalence queries in Query Learning, and that a representation class that is poly-query (resp., efficiently) learnable with membership and equivalence queries is also poly-query (resp., efficiently) elicitable with value and demand queries. They are thus able to apply more powerful results from the learning community. They consider three representation classes as examples: Polynomial representations (fully expressive and particularly succinct for valuations with complementarity), Linear-Threshold functions (representing r-of-k valuations) and valuations expressed in the XOR language (fully expressive with free disposal and succinct for valuations with substitutability). All are efficiently elicitable and the first two are poly-query elicitable.

Unlike the previously cited papers, Lahaie and Parkes (2004) also considers the incentives problem, that is, how to make sure the agents will respond truthfully to the queries. As we have seen, applying Vickrey payments at the end of a sequential elicitation process makes answering truthfully an ex-post equilibrium. But this requires eliciting extra information to determine the Vickrey payments. Lahaie and Parkes modify their definition of demand queries into *universal demand queries* which are basically $n + 1$ demand queries: one for the whole economy with n agents and one for each of the sub-economies with one agent removed. This allows the elicitor to learn the allocation functions of all $n + 1$ economies and thus to determine the Vickrey payments. Using universal demand queries does not affect the results above, except for the fact that each query requires n times as many bits to communicate. An interesting question left open in Lahaie and Parkes (2004) is that of which prices to choose in demand queries, since competitive equilibria are not unique.

5.2.3 Decision Theory

The heuristics-based and learning theory approaches described above only consider "optimal" preference elicitation in that they require that enough information be solicited so that the optimal allocation (and possibly the corresponding payments) can be determined. A decision theoretic approach, however, would allow the elicitor to trade off the quality of the decision with the costs of the elicitation. For example, even if the optimal allocation cannot be determined with the available information, asking an additional query might cost more than the gain in decision quality it will provide, and should therefore be avoided. This approach has been used for the single user preference elicitation problem in White *et al.* (1984);

Sykes and White (1991) (under non-deterministic uncertainty over preferences) and more recently in Chajewska *et al.* (2000, 2001); Boutilier (2002a) (under probabilistic uncertainty), as well as in the context of one-item auctions with price queries (Kress and Boutilier (2004), see Section 5.1.1), but has yet to be applied to the general case of multi-user preference elicitation, i.e. mechanism design.

In Boutilier (2002a), preference elicitation is modeled as a partially observable Markov decision process (POMDP), which is a Markov decision process where only part of the state space is observable. Formally, in this POMDP model, the set of states is the set of utility functions a user can have, and thus a belief state is a probability distribution over utility functions. The set of actions consists of both decisions and queries. The underlying state (the utility of the user) never changes so there are no transitions associated with a query, whereas a decision simply transitions to an additional terminating state. The reward function of the POMDP includes both the reward associated with a decision, which is the expected utility of that decision given the current belief state, and the cost associated with a query. The observation function simply models the responses of the agent to the queries. The solution, or optimal policy, for such a POMDP indicates, for all beliefs over the agent's utility, whether to elicit further information, and which query will provide the information with highest value, or to make a decision, and which decision has highest expected utility.

A POMDP formulation has several advantages over the approaches described above: it allows for more appropriate termination criteria since elicitation will stop if its cost exceeds the expected value of the information that will be elicited; it can be used with arbitrarily complex cost models, such as associating a different cost with different types of queries or even with different responses; the observation function can model noisy responses from the agents. Furthermore, a POMDP model uses probabilistic prior information on agents utilities but provides a solution for any such prior.

This approach, however, also entails severe computational issues. A preference elicitation POMDP has continuous state, action and sometimes even observation spaces. Belief states must therefore be represented using some parametric form that will either be closed under updates or easily refitted. Approximation techniques are proposed in Boutilier (2002a) but even those are computationally intensive. Extending this approach to multi-agents problems, such as a combinatorial auction would make things much worse: computing which decision has maximum expected utility (i.e., winner determination) is NP-hard, and this must be done often, including to compute the expected value of information of a query. However, all the computation involved in solving a POMDP can be done off-line and need only be done once. A POMDP model does not deal with truth-telling incentives either. The usual approach of finding Vickrey payments to provide a truthful ex-post equilibrium does not straightforwardly apply when either the elicitation process is terminated early because the cost of further information exceeds its value, or when approximation techniques are used.

6 Automated Mechanism Design

As previously discussed, classical mechanism design focuses on implementing social choice functions for all possible valuations that an agent can have. This requirement leads to two major Impossibility results: the Gibbard-Satterthwaite and Myerson-Satterthwaite theorems. One way to circumvent these negative results is to restrict the space of valuations that agents can have. In auction settings, agents are assumed to have quasi-linear utilities and the VCG mechanisms show how this restriction allows the implementation of a social welfare maximizing social choice function. This approach can be taken further: if a designer has some prior information over the agents' utilities, this should be taken advantage of in the design of the mechanism. This not only allows one to get around the Impossibility results, but even in settings when implementation is possible for a wide class of utilities, restricting this class further may sometimes allow the mechanism to achieve a higher objective value.

A classical example is the Myerson auction (Myerson (1981)). In order to maximize the seller's revenue, the Myerson auction uses agents' bids and a prior over their valuations to define each agent's *virtual valuation*, a linear transformation of the true valuation that tends to favor agents with lower valuations so as to maximize competition. The auction allocates the item to the agent with highest virtual valuation and sets the payment as the minimum valuation that agent could have and still win the auction. If the seller has non-zero valuation for the item, then no allocation is made if the highest virtual valuation is below the seller's valuation. The Myerson auction is Bayesian incentive compatible and ex-interim individually rational. For any given prior it maximizes the seller's expected revenue.

The idea of designing mechanisms tailored to specific situations using prior information has often been used in practice in settings other than maximizing the revenue of a one-item auction. In the artificial intelligence community it has been introduced as *automated mechanism design* (AMD, Conitzer and Sandholm (2002)). Conitzer and Sandholm (2002; 2003) show how this can be formulated as an optimization problem where the objective function is the expected value of the designer's objective, the variables represent the outcome function of the mechanism, and the constraints impose some concept of incentive compatibility (dominant or Bayesian) and of individual rationality (ex-post or ex-interim). This allows to find the optimal direct, full revelation mechanism (given a specific prior, in settings with a finite number of types, actions and outcomes) in polynomial time, for a fixed number of agents, by solving a linear program. Of course in most practical settings the number of outcomes, actions and types is very large, if not infinite, and further research in search algorithms specific to AMD is required.

Myerson's auction and Conitzer and Sandholm's AMD assume that the designer has *probabilistic* prior information over the agents' types. In many settings, however, this is not always realistic because such priors can be very hard or very costly to determine. Often prior information on types can be represented as upper and lower bounds on the agents' utilities for each outcome with no knowledge of how the true utilities are distributed within those bounds. This is what we call *strict uncertainty*. This has been addressed in Hyafil and Boutilier (2004) and is described in the next section.

7 Proposed Future Research

As we have seen there are still a number of significant issues in classical mechanism design that have not been fully addressed. Approaches that involve addressing these issues independently within the framework of VCG mechanisms fail when combined together. Sequential mechanisms have shown a lot of potential but much work remains to be done. Of particular interest is the investigation of *partial revelation* mechanisms. Focusing on communication complexity has several advantages. It is of course itself an important bottleneck in any setting with a large outcome space or when the utilities for these outcomes are of the same order as the communication costs. But restricting communication can, if done appropriately, also simplify the agents' valuation problem by requiring only bounds on utilities or that they only report their utility for a subset of all outcomes. Computational complexity, although no less crucial than communication complexity, should be addressed subsequently: one would have little use for a polynomial-time algorithm with an exponential number of inputs.

In previous work, we considered the problem of automatically generating partial revelation mechanisms under strict type uncertainty. Current work is focusing on a general formal model of partial revelation under various assumptions.

7.1 Previous Work

In Hyafil and Boutilier (2004), we relax the assumption of probabilistic prior information, central to Bayesian games and Bayesian AMD, to the strict uncertainty case. We introduce games of incomplete information under strict type uncertainty, which consist of the same components as a Bayesian game but with a strict, or qualitative, prior T which is a subset of the type profile space Θ. In such a setting, agents can no longer pick the strategy that maximizes their expected utility given a strategy of the others. We argue that such agents should instead attempt to minimize their regret.

Intuitively, if we fix the behavior and types of all other agents, the regret of agent i with type θ_i for playing σ_i is the loss i experiences by playing σ_i rather than acting optimally. Of course, agent i does not know the true types of the other agents. The max regret of σ_i given prior T is the most i could regret playing σ_i (against the fixed strategies of the others) should an adversary choose its opponents's types in a manner consistent with its beliefs. Finally, a minimax best response is any strategy that minimizes this worst case loss in the face of such an adversary. Note that this strategy requires a minimax optimal choice for every possible type agent i could possess. A strategy profile σ is a *minimax-regret equilibrium* if and only if σ_i is a minimax best response to σ_{-i} for all agents i. We show that a mixed strategy minimax-regret equilibrium always exists for any strict incomplete information game with finite agent, action and type space.

We then consider the problem of designing mechanisms under strict uncertainty and argue that the designer himself should be a regret minimizer: the regret of mechanism **p** relative to **q** is the maximal loss in objective value incurred by adopting **p**, allowing an adversary to choose the agents' types θ. We formulate strict AMD as an optimization problem where the designer searches for the mechanism with minimum max regret subject to the constraint that truth-telling is a minimax-equilibrium for the agents. We extend this to the case of partial revelation mechanisms, where the agents only reveal a subset in which (they claim) their true type lies. We finally show how constraint generation and constraint linearization techniques can be used to solve this (complex) optimization problem using only a sequence of linear and mixed integer programs.

All the computational issues that arise in Bayesian AMD persist under strict uncertainty and there is much work remaining to be done to make both of these approaches practical through, for instance, approximation techniques and use of structure in action, outcome and type spaces.

7.2 Current and Future Work

First let us formalize the model of partial revelation we are considering. Recall that the *full revelation* mechanism design model consists of:

- N agents, a set of outcomes, X,and of types $\Theta = \Theta_1 \times \ldots \times \Theta_N$
- Bayesian or strict prior over types: $Pr(\Theta)$ or $T \subset \Theta$
- Objective function: $f_o : \Theta \times X \to \mathbb{R}$
- Social choice function: $f : \Theta \to X$ such that $\forall \theta \in \Theta, f(\theta) \in arg\max_{x \in X} f_o(\theta, x)$

We are interested in partial revelation where, instead of revealing their full type θ_i, agents reveal a subset $\tilde{\theta}_i$ in which their true type lies. The model is therefore extended to include a (finite) partition of the type space: $\tilde{\Theta} = \tilde{\Theta}_1 \times \ldots \times \tilde{\Theta}_N$ of Θ.

Under partial revelation, directly maximizing the objective function f_o is impossible since their is still some uncertainty about agents' full types. We must therefore define a partial revelation objective function g_o that "corresponds" to the full revelation objective f_o. As always with decision making under strict uncertainty, a number of different criteria can be used (French (1986), see examples below). Any definition of g_o straight-forwardly defines a partial revelation social choice function g:

- Objective Function: $g_o : \tilde{\Theta} \times X \to \mathbb{R}$
- Social choice function: $g : \tilde{\Theta} \to X$, such that $g(\tilde{\theta}) \in arg\max_{x \in X} g_o(\tilde{\theta}, x)$

Whatever the decision criterion used to define g_o, we are interested in the following question: if f is implementable, is g implementable? We are considering three different decision criteria: maximizing the ex-post (after partial revelation) expected value of f_o, minimizing the ex-post maximum regret with respect to f_o, and maximizing the ex-post worst case value of f_o. Formally:

$$\begin{aligned}
\text{Maximizing ex-post expected objective: } g_o(\tilde{\theta}, x) &= \sum_{\theta \in \tilde{\theta}} Pr(\theta | \theta \in \tilde{\theta}) f_o(\theta, x) \\
\text{Minimizing ex-post max regret: } g_o(\tilde{\theta}, x) &= \max_{\theta \in \tilde{\theta}} \max_{x' \in X} f_o(\theta, x') - f_o(\theta, x) \\
\text{Maximizing ex-post min objective: } g_o(\tilde{\theta}, x) &\geq \min_{\theta \in \tilde{\theta}} f_o(\theta, x)
\end{aligned}$$

The partial revelation model can be applied along several independent dimensions: direct or sequential mechanisms, Bayesian or strict uncertainty, prior-independent or automatically designed mechanisms. Our current work is focusing on general (VCG-type) direct mechanisms under partial revelation. Apart from the case of maximizing ex-post expected objective, these are all prior-independent settings. Positive results on the implementability of partial revelation social choice functions in the most general case might not be obtainable so we are, for now, focusing on quasi-linear environments.

Existing work on partial revelation has been mostly unsuccessful, in large part because these approaches attempt to solve the problem optimally, that is to implement f directly without full revelation. With the notable exception of priority games (Blumrosen and Nisan (2002)), there has been little attempt to implement social choice function that provide only reasonable guarantees on the objective such as those implied by the three decision criteria above. Furthermore, such approximate partial revelation mechanisms should be able to trade off economic efficiency (as defined by the objective function) with the various costs involved with the mechanism such as communication, information revelation (privacy), valuation, and even computational costs. These costs should be taken into account within the objective function and the trade-off should be decision-theoretically optimal, as in Kress and Boutilier (2004).

Our goal is to design mechanisms within this framework that provide good incentives properties and are computationally tractable. This goal will most likely only be achievable incrementally.

- The first step is to find theoretical results on the implementability of partial revelation social choice functions, as defined in the model above.
- With such results in hand, the question is whether the partitioning of the type space can be optimized so as to maximize the guarantees provided by the mechanism on the achieved objective value, preferably in a way that minimizes each agent's valuation problem.
- The next step is to include the various costs in the objective function to obtain an optimal trade-off.
- The framework should then be extended to sequential mechanisms, which have shown great potential in reducing all types of complexity in computational mechanism design.

References

Andersson, A., Tenhunen, M., and Ygge, F. (2000). Integer programming for combinatorial auction winner determination. In *ICMAS*, pages 39–46.

Ausubel, L. (2000). An efficient dynamic auction for heterogeneous commodities. *Technical Report*.

Bichler, M. and Kalagnanam, J. (2002). Bidding languages and winner determination in multi-attribute auctions. In *IBM Research Report*.

Bikhchandani, S. and Ostroy, J. (2002). The Package Assignment Model. *Journal of Economic Theory*, pages 377–406.

Blum, A., Jackson, J., Sandholm, T., and Zinkevich, M. (2004). Preference elicitation and query learning. In *Journal of Machine Learning Research*, pages 649–667.

Blumrosen, L. and Nisan, N. (2002). Auctions with severely bounded communication. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS-02)*, pages 406–416, Vancouver.

Blumrosen, L., Nisan, N., and Segal, I. (2003). Multi-player and multi-round auctions with severely bounded communication. In *11th Annual European Symposium on Algorithms (ESA-03)*, Budapest.

Boutilier, C. (2002a). A POMDP formulation of preference elicitation problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 239–246, Edmonton.

Boutilier, C. (2002b). Solving concisely expressed combinatorial auction problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 359–366, Edmonton.

Boutilier, C. and Hoos, H. H. (2001). Bidding languages for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 1211–1217, Seattle.

Chajewska, U., Koller, D., and Parr, R. (2000). Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*.

Chajewska, U., Koller, D., and Ormoneit, D. (2001). Learning an agent's utility function by observing behavior. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*.

Conen, W. and Sandholm, T. (2001). Preference elicitation in combinatorial auctions: Extended abstract. In *Proceedings of the ACM conference on Electronic Commerce*, pages 256–259, Tampa, FL.

Conen, W. and Sandholm, T. (2002a). Differential-revelation vcg mechanism for combinatorial auctions. In *AAMAS-02 workshop on Agent-Mediated Electronic Commerce*, Bologna, Italy.

Conen, W. and Sandholm, T. (2002b). Partial-revelation vcg mechanism for combinatorial auctions. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 367–372, Edmonton.

Conitzer, V. and Sandholm, T. (2002). Complexity of mechanism design. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 103–110, Edmonton, Alberta, Canada.

Demange, G., Gale, D., and Sotomayor, M. (1986a). Multi-item auctions. *Journal of Political Economy*, **94**(4), 863–872.

Demange, G., Gale, D., and Sotomayor, M. (1986b). Multi-item auctions. *Journal of Political Economy*, pages 863–872.

French, S. (1986). *Decision Theory*. Halsted Press, New York.

Fujishima, Y., Leyton-Brown, K., and Shoham, Y. (1999). Taming the computational complexity of combinatorial auctions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553, Stockholm.

Grigorieva, E., Herings, J.-J. P., Müller, R., and Vermeulen, D. (2002). The private value single item bisection auction. Research Memorandum 51, METEOR, Maastricht.

Gul, F. and Stachetti, E. (2000). The English auction with differentiated commodities. *Journal of Economic Theory*, pages 66–95.

Hastad, J. (1999). Clique is hard to approximate within $n^{(1-\epsilon)}$. In *Acta Mathematica*, volume 182, pages 105–142.

Holzman, R., Kfir-Dahav, N., Monderer, D., and Tennenholtz, M. (2003). Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*.

Hoos, H. H. and Boutilier, C. (2000). Solving combinatorial auctions using stochastic local search. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 22–29, Austin, TX.

Hudson, B. and Sandholm, T. (2004). Effectiveness of query types and policies for preference elicitation in combinatorial auctions. In *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS*, pages 386–393, New York.

Hyafil, N. and Boutilier, C. (2004). Regret Minimizing Equilibria and Mechanisms for Games with Strict Type Uncertainty. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, Banff, Alberta, Canada.

Keeney, R. L. and Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Wiley, New York.

Kelso, A. and Crawford, V. (1982). Job matching, coalition formation, and gross substitues. *Econometrica*, pages 1483–1504.

Kress, A. and Boutilier, C. (2004). A Study of Limited-Precision, Incremental Elicitation in Auctions. In *Proceedings of the third international joint conference of autonomous agents and multi-agent systems (AAMAS 04)*, pages 1344–1345, New York, NY, USA.

Lahaie, S. M. and Parkes, D. C. (2004). Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 180–188. ACM Press.

Larson, K. and Sandholm, T. (2001a). Computationally limited agents in auctions. In *Workshop on Agent-based Approaches to B2B, Autonomous Agents 2001*, Montreal, Quebec, Canada.

Larson, K. and Sandholm, T. (2001b). Costly valuation computation in auctions. In *In Proceedings of the Eighth Conference of Theoretical Aspects of Knowledge and Rationality (TARK VIII)*, Sienna, Italy.

Larson, K. and Sandholm, T. (2003). Miscomputing ratio: the social cost of selfish computing. In *In Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, Melbourne, Australia.

Larson, K. and Sandholm, T. (2004). Strategic deliberation and truthful revelation: an impossibility result. In *In the proceedings of ACM Conference on Electronic Commerce (EC-2004)*.

Lehman, D., O'Callaghan, L., and Shoham, Y. (2002). Truth revelation in approximately efficient combinatorial auctions. In *Journal of the ACM*, pages 577–602.

Leyton-Brown, K., Pearson, M., and Shoham, Y. (2000). Towards a universal test suite for combinatorial auction algorithms. In *ACM Conference on Electronic Commerce*.

MacKie-Mason, J. K. and Varian, H. (1994). Generalized vickrey auctions. In *Working paper, Un. of Michigan, July 1994*.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press, New York.

Myerson, R. B. (1981). Optimal auction design. In *Mathematics of Operations Research*, pages 58–73.

Nisan, N. (2000). Bidding and allocations in combinatorial auctions. In *ACM Conference on Electronic Commerce (EC-2000)*, pages 1–12, Minneapolis, MI.

Nisan, N. and Ronen, A. (2000). Computationally feasible VCG mechanisms. In *ACM Conference on Electronic Commerce (EC-2000)*, pages 242–252, Minneapolis, MI.

Nisan, N. and Segal, I. (2003). The communication requirements of efficient allocations and supporting lindahl prices. In *Working paper*.

Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press, Cambridge.

Parkes, D. C. (2001). An iterative generalized Vickrey auction: Strategy-proofness without complete revelation. In *Proc. AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents*, pages 78–87. AAAI Press.

Parkes, D. C. (2004). Auction design with costly preference elicitation. *Annals of Mathematics and AI*. Special Issue on the Foundations of Electronic Commerce, Forthcoming.

Parkes, D. C. and Kalagnanam, J. (2004). Iterative Multiattribute Vickrey Auctions. *Management Science*. Special Issue on Electronic Markets.

Parkes, D. C. and Ungar, L. H. (2000a). Iterative combinatorial auctions: Theory and practice. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 74–81, Austin, TX.

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

Ronen, A. (2001). Mechanism design with incomplete languages. In *Proceedings of the Third ACM Conference on Electronic Commerce*.

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

Sandholm, T. (1999). Algorithm for optimal winner determination in combinatorial auctions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

Sandholm, T. (2003). Automated mechanism design: A new application area for search algorithms. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP-03)*, Kinsale, Ireland.

Sandholm, T., Suri, S., Gilpin, A., and Levine, D. (2001). CABOB: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 1102–1108, Seattle.

Sanghvi, S. and Parkes, D. C. (2004). Hard-to-manipulate combinatorial auctions. Technical report, Harvard University.

Santi, P., Conitzer, V., and Sandholm, T. (2004). Towards a Characterization of Polynomial Preference Elicitation with Value Queries in Combinatorial Auctions. In *Seventeenth Conference on Learning Theory (COLT)*, pages 1–16.

Sunderam, A. V. and Parkes, D. C. (2003). Preference elicitation in proxied multiattribute auctions. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, pages 214–215.

Sykes, E. A. and White, III, C. C. (1991). Multiobjective intelligent computer-aided design. *IEEE Transactions on Systems, Man and Cybernetics*, **21**(6), 1498–1511.

Tennenholtz, M. (2000). Some tractable combinatorial auctions. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ.

White, III, C. C., Sage, A. P., and Dozono, S. (1984). A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man and Cybernetics*, **14**(2), 223–229.

Wurman, P. and Wellman, M. (2000). Akba: A progressive, anonymous-price combinatorial auction. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 21–29, Minneapolis, MN.

Zinkevich, M., Blum, A., and Sandholm, T. (2003). On polynomial-time preference elicitation with value-queries. In *Proceedings of the Fourth ACM Conference on Electronic Commerce*, San Diego, CA.