Mechanism Design with Incomplete Languages

Amir Ronen*
Department of Computer Science
Stanford University
(amirr@Robotics.Stanford.EDU)

Abstract

A major achievement of mechanism design theory is the family of truthful mechanisms often called VCG (named after Vickrey, Clarke and Groves). Although these mechanisms have many appealing properties, their essential intractability prevents them from being applied to complex problems like combinatorial auctions. In particular, VCG mechanisms require the agents to **fully describe** their valuation functions to the mechanism. Such a description may require exponential size and thus be infeasible for the agents.

A natural approach for this problem is to introduce an intermediate language for the description of the valuations. Such a language must be succinct to both the agents and the mechanism. Unfortunately, the resulting mechanisms are neither truthful nor do they satisfy individual rationality.

This paper suggests a general method for overcoming this difficulty. Given an intermediate language and an algorithm for computing the results, we propose three different mechanisms, each more powerful than its predecessor, but also more time consuming. Under reasonable assumptions, the results of our mechanisms are **at least as good** as the results of the algorithm on the actual valuations. All of our mechanisms have polynomial computational time and satisfy individual rationality.

1 Introduction

1.1 Motivation

The theory of mechanism design may be described as studying the design of protocols under the assumption that the participants behave according

*This research was supported by Darpa grants number F30602-98-C-0214 and F30602-00-2-0598.

to their own goals and preferences and not necessarily as instructed by the protocol. The canonical mechanism design problem can be described as follows: A set of rational agents need to collaboratively choose an outcome o from a finite set O of possibilities. Each agent i has a privately known valuation function $v^i : O \to R$ quantifying the agent's benefit from each possible outcome. The agents are supposed to report their valuation functions $v^i(\cdot)$ to some centralized mechanism that chooses an outcome o that maximizes the total welfare $\sum_i v^i(o)$. The main difficulty is that agents may choose not to reveal their true valuations but rather report carefully designed lies in an attempt to influence the outcome to their liking. The tool that the mechanism uses to motivate the agents to reveal the truth is monetary payments. These payments are to be designed in a way that ensures that rational agents always reveal their true valuations – making the mechanism, so called, incentive compatible or truthful. To date there is only one general technique known for designing such a payment structure, sometimes called the generalized Vickrey auction [21], the Clarke pivot rule [1] the Groves mechanism [5], or, as we will, VCG. In certain senses this payment structure is unique [4, 17].

Although VCG mechanisms have many appealing properties, their **intractibility** prevents them from being applied to complex problems like combinatorial auctions. This intractability is twofold: Firstly, VCG mechanisms require the agents to fully describe their valuation functions. Secondly, it requires the mechanism to find the optimal allocation.

The problem of combinatorial auctions (CA) is an important example of a mechanism design problem. In CA, the designer would like to auction a set S of items (e.g. radio spectra licenses) among a group of agents who desire them. As items may be substitutes (e.g. two licenses in the same place) or complementary (e.g. licenses in two neighboring states) the valuation of each agent may have a complex structure. A formal definition of the problem can be found in section 2.1.

Consider a VCG mechanism for CA: The mechanism first asks each agent to declare her valuation function, i.e. to report a function $w^i : 2^S \rightarrow R_+$. It then computes the optimal allocation and the payments of each agent.

Such a mechanism is clearly intractable. Firstly, finding the optimal allocation is NP-hard even to approximate. Secondly, the mechanism relies on the agents' ability to describe their valuations in a way which is succinct to its allocation algorithm. This ability cannot be taken for granted. For example a naive solution will require each agent to report a vector of $2^{|S|} - 1$ numbers to the mechanism. This of course is not feasible unless the number of items is very small. On the other extreme the designer can ask the agents to submit *oracles*, i.e. programs that return for every set s their valuation $v^i(s)$. However, it is not difficult to see that in order to find the optimal allocation or even a reasonable one, the allocation algorithm must query these oracles an exponential number of times. The natural solution for this problem is to introduce the notion of a *bidding language*. Such a language should enable the agents to efficiently represent or at least to approximate their valuations, but should also allow the allocation algorithm to compute the desired allocation in polynomial time. Hopefully such a language will capture most "real life" valuations. Various bidding languages were proposed in recent years. The interested reader is pointed to [11].

The drawback of this approach is that there are always valuation functions which are impossible to represent in polynomial-time. We therefore call such languages incomplete. Since VCG mechanisms with incomplete languages are not optimal, the impossibility results of [13] imply that they *cannot* be truthful! In other words, instead of describing their true valuation according to the designer's instructions, agents may have incentive to misreport. Therefore, there is no guarantee, even when the agents are rational, that the mechanism will find a reasonable allocation. Moreover, such mechanisms do not even guarantee individual rationality. That is, there are cases where truthful agents will pay for their allocated sets more than their actual valuations for them.

Our goal in this paper is to prevent these phenomena.

1.2 This work

This paper proposes a general method for overcoming the non truthfulness of VCG mechanisms with incomplete languages. We first introduce the notions of oracles[1], descriptions and consistency checkers in the context of VCG mechanisms. Oracles are programs that represent the agents' valuations. They are used by the mechanism to measure the agents' welfare. A consistency checker is a function that checks whether an agent's description, which is given in the intermediate language, is consistent with her oracle. These additions to the VCG method still do not suffice to guarantee its truthfulness.

We then describe three mechanisms which guarantee that under reasonable assumptions, truth-telling is *the* rational strategy for the agents. Each mechanism is more powerful but also more time consuming than its predecessor. All of our mechanisms have polynomial computational time.

Following [13] we adopt the concept of feasibly dominant actions (FDAs). Informally speaking, we assume that the agents choose their actions (strategies) according to their strategic knowledge. We say that an action is *feasibly domiant* if the agent is not aware of any circumstances where another strategy is better for her. It was argued in [13] that when feasibly dominant actions are available for the agent, it is irrational for her not to choose one of them. It was also shown in [13] that if the payment of a non-optimal mechanism is calculated according to the VCG formula, the existence of FDAs *must* rely on further assumptions on the agent's knowledge. Our mecha-

[1]Some advantages of using oracles were discussed in [19]

nisms guarantee that, under such reasonable assumptions, truth-telling is indeed an FDA. Each of them handles a more general form of knowledge than its predecessor (i.e. more sophisticated agents).

When the agents are truthful, the result of our mechanisms is **at least as good** as the result of the allocation algorithm on the truthfully reported descriptions. Our mechanisms also satisfy individual rationality.

Note that our method does not make any assumptions on the algorithm or the bidding language. The designer needs to design an intermediate language, a consistency checker and an allocation algorithm such that, when the agents prepare their descriptions according to her instructions, the overall result is good. She then gets the mechanism for free.

For simplicity we prove all our theorems directly for the combinatorial auction problem. Our results however are much more general and can be applied to any VCG, weighted VCG or compensation and bonus [14] mechanism.

1.3 Related work

Non optimal VCG mechanisms were first studied in [13]. This paper discusses VCG mechanisms where the optimal algorithm is replaced by a poly-time approximation or heuristic. This paper shows that mechanisms constructed this way *cannot* be truthful. It then proposes a general way of dealing with this non-truthfulness using a certain form of appeal functions.

The problem of combinatorial auctions has been studied by several researchers in recent years. A comprehensive survey of various aspects of this problem can be found in [2]. In particular, various bidding languages [3, 7, 20] and restrictions on the classes of bids that can be submitted (e.g. [6]) were proposed. A comparative study of some of these languages can be found in [11].

An alternative approach to the one that is taken here is to consider mechanisms where the agents are not required to declare their valuation functions (non-revelation mechanisms). Examples of such mechanisms are the simultaneous ascending auction [10] and iBundle [16]. The efficiency of these auctions however is dependent on strong assumptions on the agents' behaviour. They are also specifically designed to address the combinatorial auction problem.

Finally, there is an extensive literature in the field of mechanism design. An introduction can be found in [8, chapter 23] and [15, chapter 10]

Organization of this paper: The rest of the paper is organized as follows: Section 2 formally defines combinatorial auctions and VCG mechanisms for CA and explains their intractability. Section 3 provides an example of a VCG mechanism with incomplete language and demonstrates the drawbacks of such mechanisms. Section 4 defines our most basic mechanism, describes the main concepts of [13] and shows that under reasonable assumptions on the agents' knowledge, truth-telling is an FDA. Sections 4 to 6 define extended versions of this mechanism and prove their basic properties. Section 7 discusses additional implementation issues and section 8 concludes the paper.

2 Preliminaries

2.1 Combinatorial auctions (CA)

The problem of combinatorial auctions (CA) has been extensively studied in recent years (see e.g. [7] [20] [3] [6] [11]). The importance of this problem is twofold. Firstly, several important applications rely on it (e.g. the FCC auction [9]). Secondly, it is a generalization of many other problems of interest, in particular in the field of electronic commerce. A recent survey of various aspects of this problem can be found in [2]. For simplicity we prove all our theorems directly for this problem.

The problem: A seller wishes to sell a set S of items (radio spectra licenses, electronic devices, etc.) to a group of n agents who desire them. Each agent i has, for every subset $s \subseteq S$ of the items, a non-negative number $v^i(s)$ that represents how much s is worth for her. The function $v^i(.)$ is called the agent's *valuation* or *type*. We assume that $v^i(.)$ is **privately** known to the agent. Given a (possibly partial) allocation $s = (s^1, \ldots, s^n)$ we shall define the *total welfare* of the agents as $g = \sum_i v^i(s)$. In this paper we will be interested in mechanisms (protocols) which are designed to *maximize* the total welfare. This goal is justified in many settings. There is also a basic correlation between maximizing welfare and maximizing the seller's revenue. Solving the problem without monetary transfers is impossible (see a discussion at [8, chapter 23]). We assume that the mechanism can ask for payment from the agents and that the overall *utility* of each agent i is $u^i = v^i(s) + p^i$ where s denotes the chosen allocation and p^i the amount of currency that the mechanism pays to the agent[2]. In an auction, p^i will be non-positive. This utility is what each *agent* tries to maximize.

For the sake of the example we take some standard additional assumptions on the type space of the agents:

No externalities The valuation of each agent depends only on the items allocated to her. I.e. $\{v^i(s^i) | s \subseteq S)\}$ completely represents the agent's valuation.

Free disposal Items have non-negative values. I.e if $s \subseteq t$ then $v^i(s) \leq v^i(t)$.

Normalization $v^i(\phi) = 0$.

Note that the problem allows items to be complementary, i.e. $v^i(S \bigcup T) \geq v^i(S) + v^i(T)$ or substitutes, i.e. $v^i(S \bigcup T) \leq v^i(S) + v^i(T)$

[2] This is called the quasi-linearity assumption.

(S, T disjointed). For example an agent may be willing to pay \$200 for a TV set, \$150 for a VCR, \$450 for both and only \$200 for two VCRs. The structure of the valuation functions might therefore be complex. The problem of finding an optimal allocation is equivalent to set-packing and is NP-hard even to approximate within any reasonable factor.

Note that the valuation functions are not known to the mechanism in advance. Moreover, if the mechanism is not carefully designed, the agents will have an incentive to **manipulate** it for their own self interest. Such manipulations might severely damage the efficiency of the mechanism. In mechanism design problems the agents are assumed to be *rational* in a game theoretic sense. They choose strategies which are good for them and not necessarily act as instructed. The goal of the designer is to design a mechanism (protocol) that produces good results under this assumption. Comprehensive surveys of mechanism design theory can be found in [15, chapter 10] [8, chapter 23].

In order to handle complex problems like combinatorial auctions the mechanism needs to address the following issues:

- Agents' valuations might be complex to express.
- The allocation and payments might be hard to compute.
- The mechanism needs to be designed to find good allocations even though the agents follow their own self interest.

Let us summarize our notations and terminology regarding this problem.

Notations: We shall denote the whole set of items by S and a (possibly partial) allocation by $s = (s^1, \dots, s^n)$. Note that the s^is are disjointed. We denote the type of agent i by v^i and the group's type by $v = (v^1, \dots, v^n)$. Let p^i denote the amount of currency that the mechanism pays to each agent i and u^i the agent's utility. Given an allocation s and a type v we denote by

$g_s(v)$ the welfare $\sum_i v^i(s^i)$. Finally we shall use the following vectorial notation: given a vector $a = (a^1, \ldots, a^n)$ we let $a^{-i} = (a^1, \ldots, a^{i-1}, a^{i+1}, \ldots, a^n)$ and (b^i, a^{-i}) denote the vector $(a^1, \ldots, a^{i-1}, b^i, a^{i+1}, \ldots, a^n)$.

2.2 VCG mechanisms for CA

One of the major achievements of mechanism design theory is the VCG method for constructing truthful mechanisms. In this subsection we briefly describe these mechanisms for CA and discuss some of their properties.

The simplest kind of mechanisms are protocols (called revelation mechanisms) where the agents are simply required to (privately) report their types to the mechanism. According to these declarations the mechanism computes the allocation and the payments. Note that agents may **lie** if it is beneficial for them. Such a mechanism can be denoted by a pair $m = (k(w), p(w))$ where k denotes the allocation function, p the payment function and w the agents' declaration .

Definition 1 (truthful mechanism) *A revelation mechanism is called* truthful *if truth-telling is a dominant strategy for all agents. I.e. if lying to the mechanism can never be more beneficial than declaring v^i.*

VCG mechanism are a special kind of revelation mechanisms.

Definition 2 (VCG mechanism) *A* VCG mechanism *for CA is a revelation mechanism $m = (k(w), p(w))$ such that:*

- *The mechanism chooses an allocation $s = k(w)$ that maximizes the total welfare $g_s(w)$ according to the declaration w.*
- *The payment is calculated according to the* VCG formula*: $p^i(w) = \sum_{j \neq i} w^j(s)) + h^i(w^{-i})$ ($h^i(.)$ can be any real function of w^{-i}).*

Theorem 2.1 *([5]) A VCG mechanism is truthful.*

Proof: Assume by contradiction that the mechanism is not truthful. Then there exists an agent i of type v^i, a type declaration w^{-i} for the other agents, and $w^i \neq v^i$ such that $v^i(k((v^i, w^{-i}))) + p^i((v^i, w^{-i})) + h^i(w^{-i}) < v^i(k((w^i, w^{-i}))) + p^i((w^i, w^{-i})) + h^i(w^{-i})$. Let $s = k((v^i, w^{-i}))$ denote the chosen allocation when the agent is truthful and let $s' = k((w^i, w^{-i}))$. The above inequality implies that $g_s((v^i, w^{-i})) < g_{s'}((v^i, w^{-i}))$. This contradicts the *optimality* of $k(.)$. $\square$

Rational agents will therefore reveal their true type to the mechanism. Thus, when agents are rational the mechanism will result in the *optimal* allocation!

Note that the main trick of this method is to identify the utility of truthful agents with the declared total welfare. Similar techniques were introduced in [14] for handling different type of problems. The results presented here are applicable to their methods as well.

Another desirable property of mechanisms is called *individual rationality*. This means that the utility of a truthful agent is guaranteed to be non-negative. A special kind of VCG mechanism called Clarke's mechanism [1] can guarantee this property. It also guarantees that the payment of agents who are not allocated any object is zero. It does so by setting $h^i = -\sum_{j \neq i} k(w^{-i})$ where $k(w^{-i})$ denotes the result of the algorithm when agent i is "ignored". Until section 7 we shall only be interested in truthfulness. Thus, for simplicity we can assume that $h^i(w^{-i}) \equiv 0$.

It is worth notifying that weighted VCG mechanisms are possible as well (see e.g. [17] [14]). Also the designer can impose her own preferences by "pretending" to be one of the agents. To date VCG is the **only** general known method for the construction of truthful mechanisms. There is also some evidence [17] that other methods are generally impossible.

2.3 The intractability of VCG mechanisms

Although VCG mechanisms have many desirable properties, their essential intractability prevents them from being used for complex problems like CA. This intractability is twofold: VCG mechanisms require the agents to **fully describe** their valuation functions and require the mechanism to find optimal allocations.

The second aspect has been extensively discussed in [13]. This paper discusses VCG mechanisms where the optimal algorithm is replaced by a poly-time approximation or heuristics. It shows that mechanisms which are constructed in this way *cannot* be truthful. The paper proposes a method to overcome this non-truthfulness. It suggests a bounded rationality variant of truthfulness called feasible truthfulness and shows that under reasonable assumptions there is a general way of constructing poly-time feasible truthful mechanisms.

An even more fundamental obstacle on the way to the application of VCG mechanisms (and revelation mechanisms in general) to complex problems is the fact that the agents are required to describe their valuation functions to the mechanism. Consider for example a VCG mechanism for CA. One natural way in which an agent can describe her valuation function to the mechanism is by reporting a vector of numbers denoting her valuation for every possible combination of items. This however is infeasible unless the number of items is very small as it will require a vector of size $2^{|S|} - 1$. On the other extreme, the designer can ask the agent to construct an *oracle*, i.e. a program that returns for every set s the agent's valuation $v^i(s)$. However it is not difficult to see that in order to find the best allocation or even a reasonable one, the algorithm needs to query the oracle an exponential number of times.

The natural solution for this problem is to introduce the notion of a *bidding language* (see e.g. [11]) – a language that will enable agents to efficiently represent or at least approximate their valuations but will also allow the mechanism's algorithm to compute the desired allocation in polynomial time. Hopefully such a language will capture most "real life" valuations. In addition the designer must provide the agents with *instructions* of how to construct these descriptions from their actual valuations. Given such a language L we can define VCG mechanisms as before. The bidding language and allocation algorithm must be constructed in a way that when the agents follow the designer's instructions, the results will be good (heuristically, within a certain factor from the optimum etc.)

The problem with this approach is that there are always valuation functions which are *impossible* to represent in polynomial-time. We therefore call such languages incomplete. As such a mechanism is not optimal, the impossibility results in [13] imply that VCG mechanisms with incomplete languages *cannot* be truthful! In other words, agents may have incentives not to follow the designer's instructions. Therefore there is no guarantee, even when the agents are rational, that the overall results will be good. Moreover, such mechanisms do not even guarantee individual rationality. That is, there are cases where truthful agents will pay for their allocated sets more than their actual valuations for them.

In this paper we propose a general method for overcoming this non-truthfulness. Our solution is in the same spirit of [13]. However several additional steps are needed to guarantee the good game theoretical properties of the resulting mechanisms.

3 Example VCG with OR bids

In this section we describe a simple example for a VCG mechanism with an incomplete bidding language. We shall use this example throughout

	A	B	AB
Agent1:	1	1	1.25 (2)
Agent2:	0.8	0.8	1.2 (1.6)

Figure 1: Type matrix for the OR example

the paper. We first describe the language and the mechanism. Then we analyze what strategies rational agents might choose when participating in it. Note that our language is less expressive than what we expect from real life mechanisms. We will demonstrate that even with such a language it is possible to construct mechanisms where truth-telling is *the* rational strategy.

Following [11] we define an *atomic bid* to be a pair (s, p) where $s \subseteq S$ is a set of items and p is a price. The semantic of such a bid is "my maximum willingness to pay for s is p". A description in this language consists of a *polynomial* number of such pairs. Given such a description (s_j, p_j) we can define, for every set s, the price p_s to be the maximal[3] sum of p_js such that $s_j \subseteq s$ are disjointed: $\max\{\sum_j p_j | (s_j \subseteq s) \text{ and } \forall j \neq k, s_j \bigcap s_k = \phi\}$. This so called OR language was used in [20].

Proposition 3.1 *[11] OR bids can represent only super-additive valuation functions.*

□

The OR language therefore assumes that if an agent is willing to pay up to P_A for item A and P_B for item B, then she is willing to pay at least $(P_A + P_B)$ for both.

Consider now the following (toy) example of a VCG mechanism: There are only two items A and B. As shown in figure 3, the type of Agent1 is $(1, 1, 1.25)$ and of Agent2 is $(0.8, 0.8, 1.2)$.

[3] For the sake of the example we ignore the fact that computing this maximum might be NP-hard.

Suppose that the designer instructs the agents to submit their true valuation for every singleton. In this case we can define a description $d^i = \{(s_j, p_j)\}$ as truthful if for every item j, $p_j = v^i(j)$. In other words, such a description was prepared according to the designer's instructions. Consider a VCG mechanism with this language. After the descriptions are reported, the mechanism allocates the items optimaly (according to the descriptions but not to the actual allocations!). It then calculates the payments according to the VCG formula. We assume that the designer has a small reserved price for each item, so objects which are not desired by the agents are not allocated.

In the example, when both agents are truthful, the mechanism will assign the valuation in brackets to the set AB (see figure 3). The mechanism in this case will allocate both items to Agent1 resulting in a utility of $u^i = 1.25$ for each agent (recall that we assume the simplified form where $h^i \equiv 0$). The optimal allocation will allocate to each agent one item, resulting in a welfare of 1.8.

The above mechanism is **not** truthful. For example if Agent1 "gives up" item B and declares $(1, 0, 1.25)$ while Agent2's declaration remains the same, it will cause the algorithm to produce the optimal result and therefore will increase Agent1's utility to 1.8! The same is true for Agent2. On the other hand if both agents are "giving up" the same item, only one item will be allocated (to Agent1). This will result in a welfare of only 1.0. We shall call a declaration where the agent reports a 0 value on one of the items *singleton concession*. Another reasonable strategy for an agents is to find a description which will bring the mechanism's interpretation as close as possible to her actual valuation. Formally we define the l_∞-*approximation* of $v^i(.)$ to be the description that minimizes $\max_s |v^i(s) - d^i(s)|$[4]. Such a

[4]For the sake of the example we ignore the fact that calculating such a description might be NP-hard.

description for Agent1 is $\{2/3, 2/3, 4/3\}$. Note that the worthwhileness of such declarations is highly dependent on the declarations of the others. There are cases where such declarations will considerably improve the result of the algorithm and therefore will increase the agent's utility. On the other hand there are many cases where such designated "lies" will severely damage the total welfare and henceforth the agent's utility.

Note that the Clarke version of the above mechanism does **not** satisfy individual rationality. For example, if both agents are truthful, Agent1 gets both items, but pays 1.6, thereby loosing 0.35.

In this paper we will try to prevent these bad phenomena from happening.

4 Mechanism1

In this section we describe our first and most basic mechanism. We first describe the building blocks of the mechanism – oracles, descriptions and consistency checkers. Then we define the mechanism and formulate its basic properties. Finally we show that under reasonable assumptions truth-telling is *the* rational strategy for the agents.

We start with a formal definition adopted from [13] of computationally bounded algorithms[5].

Definition 3 (algorithm of degree d) *Let n denote the number of agents. We say that a function F is* of degree d *if its running time is bounded by some polynomial of degree d of n.*

Our mechanism fixes a constant $c = O(n^d)$ and terminates each function that runs more than c time units (see section 7 for more details).

[5]There are several alternative definitions. This one simplifies the formalization of the results.

4.1 Oracles and valid descriptions

All the mechanisms described in this paper ask the agents to prepare oracles that represent their valuation functions. These oracles are queried by the mechanisms in order to measure the total welfare. Formally:

Definition 4 (oracle) *An* oracle *is a function* $w : 2^S \to R_+$*. It is called* truthful *for agent* i *if* $w^i(s) = v^i(s)$ *for every set* s*.*

We shall assume that agents are capable of preparing such oracles[6]. We also assume that all the oracles are of degree d.

As mentioned earlier, it is hard for allocation algorithms to work with oracles. We assume that the allocation algorithm accepts as input descriptions in some bidding language (e.g. the OR language) and ask the agents to prepare such descriptions. A consistency checker verifies that the agents' descriptions are consistent with their oracles.

Definition 5 (valid description) *A* consistency checker *is a function* $\psi(w, d)$ *such that:*

- $\psi(w, d)$ *gets an oracle* w *and a description* d *in the bidding language and returns a "corrected" oracle* w'*.*
- *for every oracle* w *there exists at least one description* d *such that* $w = \psi(w, d)$*. Such "fixpoint" descriptions are called* valid*.*

Semantically, a valid description was prepared according to the designer's instructions. Since the mechanism can always use the "corrected" oracle w' we shall assume that agents' descriptions are valid. We also assume that a consistency checker of degree d is available to the designer and that given a

[6]The tools which must be provided by the designer in order to make this assumption realistic are not discussed in this paper.

declaration d, the designer can compute an oracle w_d such that d is a valid description of w_d. We say that an agent's description is truthful if it is a valid description of a truthful oracle.

In the OR language for example we can define $w'(s) = p$ for every atomic bid (s, p) in the description. Creating an oracle w from a description d such that d is valid is straight forward.

4.2 Appeal functions

Another basic building block of our mechanism is the notion of appeal functions. This is a modification of the appeals that were introduced in [13]. Intuitively an appeal function lets an agent incorporate her own *knowledge* about the algorithm into the mechanism. The idea is that instead of declaring a falsified type, the agent can follow the designer's instructions and ask the mechanism to *check* whether the false description would have lead to better results. The mechanism will then choose the better of these two possibilities leveraging both the agent's utility and the total welfare.

Definition 6 (appeal) *An* appeal *function gets as input the agents' oracles and valid descriptions and returns a tuple of alternative descriptions. I.e. it is of the form: $l(w^1, \ldots, w^n, d^1, \ldots, d^n) = (d'^1, \ldots, d'^n)$ where d^i is a valid description of w^i.*

Note that the d'^is do not have to be valid. The semantics of an appeal l is: "when the agents' type is $w = (w^1, \ldots, w^n)$ and is described by $(d^1, \ldots, d^n)$, I believe that the output algorithm k produces a better result if it is given d' instead of the actual description d".

We assume that all appeal functions are of degree d for some reasonable value of d. In section 7 we will discuss ways to enforce such a limit.

In our OR example (section 3) an appeal for Agent1 might try to give up one of the items (i.e. perform a singleton concession) or try to give up item A for herself and B for Agent2 etc.

The actual implementation of the appeal functions is discussed in subsection 7.

4.3 Mechanism1

We can now define our first mechanism.

Definition 7 (mechanism1) *Given an allocation algorithm $k(d)$, and a consistency checker for the bidding language we define* mechanism1 *as follows:*

1. *Each agent submits to the mechanism:*
   - *An oracle $w^i(.)$.*
   - *A (valid) description d^i.*
   - *An appeal function $l^i(.)$.*
2. *Let $w = (w^1, \ldots, w^n)$, $d = (d^1, \ldots, d^n)$. The mechanism computes the allocations $k(d), k(l^1(w, d)), \ldots, k(l^n(w, d))$ and chooses among these allocations the one that maximizes the total welfare (according to w!). In other words, the mechanism tries all the appeals and chooses the one that yields the best result.*
3. *Let $\hat{s}$ denote the chosen allocation. The mechanism calculates the payments according to the VCG formula: $p^i = \sum_{j \neq i} w^j(\hat{s}) + h^i(w^{-i}, d^{-i}, l^{-i})$ ($h^i(.)$ can be any real function).*

Note that $h^i(.)$ is independent of agent i. Until section 7 we simply assume that it is always zero. Note also that we do not require the allocation algorithm $k(.)$ to be optimal. It can be *any* polynomial time approximation or heuristic.

An action (strategy) in mechanism1 is a triplet (w^i, d^i, l^i). We say that such an action is *truthful* if w^i is truthful. The following two observations are key properties of the mechanism:

Proposition 4.1 *Consider mechanism1 with an allocation algorithm $k(.)$. Let $d = (d^1, \ldots, d^n)$ denote the agents' descriptions. If all the agents are truth-telling, the allocation chosen by the mechanism is* **at least as good** *as $k(d)$.*

$\square$

Proposition 4.2 *If the allocation algorithm k, the appeal functions, oracles and consistency checkers are of degree d, then the mechanism is of degree $d + 2$.*

$\square$

Let $\hat{s}$ denote the chosen allocation. Let $\tilde{v} = (v^i, w^{-i})$. Since we assume that $h^i() = 0$, the utility of agent i equals $g_{\hat{s}}(\tilde{v})$ – the total welfare when the allocation is $\hat{s}$ and the type is $\tilde{v}$. Lying to the mechanism, i.e. submitting an oracle $w^i \neq v^i$, is thus beneficial for the agent *only* if it causes the mechanism to compute a better result (relatively to $\tilde{v}$). (For a more comprehensive discussion see [13].) Note that when an agent lies to the mechanism, she may not only cause damage to the algorithm's result, but may also cause the mechanism to prefer the wrong allocation on the second stage. Thus, an agent needs to have a good *reason* for lying to the mechanism.

We will show that under reasonable assumptions on the agents, truth-telling is *the* rational strategy for the agents. Thus, when the agents are rational, the result of the mechanism is at least as good as the result of the allocation algorithm on the truthful descriptions.

4.4 An example

Consider the OR example of section 3. Suppose that Agent1 notices that usually the result of the algorithm improves when she is giving up item A. In a VCG mechanism the agent may be tempted to misreport in order to increase the total welfare and henceforth her own utility. In many cases however this will cause damage to the overall welfare and henceforth to Agent1. In our mechanism Agent1 can, instead of lying, declare her **true** type to the mechanism and ask it to **check** whether such a lie would have been helpful. If so, it prefers the result that was obtained by "lying". Otherwise, the mechanism prefers the result of the algorithm on the truthful description and thus prevents the damage that would have been caused by the lie. This form of appeal functions provides the agents with a lot of power. Suppose, for example, that Agent1 notices that the result improves if she gives up item A while Agent2 is giving up item B. As before, the agent can ask the mechanism to check whether such a transformation of the input would have improve the overall result.

We note that not every knowledge of the agent about the allocation algorithm $k(.)$ can be exploited in this mechanism. Suppose that Agent1 notices that when both agents submit l_∞-approximations of their valuations the overall result improves. However, as she is given an oracle for v^2, she cannot compute Agent2's approximation as it requires her to query the oracle for every possible subset. Therefore, she cannot exploit her knowledge about the algorithm. Such phenomena is problematic and do not occur in the setting of [13].

4.5 When is it rational to tell the truth to the mechanism?

It was shown in [13] that even with full descriptions available, non-optimal VCG mechanisms cannot be truthful (unless they produce unreasonable results). That paper introduces a bounded rationality variant of the concept of dominant strategies called feasible dominance and shows that under reasonable assumptions truth-telling is feasibly dominant for the agents. This paper follows this pattern. In this section we first describe the basic concepts of [13]. We then consider mechanis1 and analyze the conditions under which truth-telling is feasibly dominant for the agents.

4.5.1 Feasibly dominant actions (FDAs)

In this section we briefly describe the main concepts of [13]. The reader is referred to this paper for a more comprehensive discussion.

Notations: We denote the action (strategy) space of agent i by A^i. Given a tuple $a = (a^1, \ldots, a^n)$ of actions chosen by the agents, we denote the utility of agent i by $u^i(a)$.
In mechanism1 an action for the agent is a triplet (w^i, d^i, l^i).

In classical game theory, given the actions of the other agents a^{-i}, the agent is (implicitly) assumed to be capable of responding by the optimal a^i. As the action space is typically very complex, this assumption is not natural in many real-life situations. The concept of feasibly dominant actions reformulates the concept of dominant actions under the assumption that the agent has only a limited capability of computing her response. It is meant to be used in the context of revelation games.

Definition 8 (strategic knowledge) Strategic knowledge *(or* response function*) of agent i is a partial function $b^i : A^{-i} \to A^i$.*

Knowledge is a function by which the agent describes (for herself!) how she would like to respond to any given situation. The semantics of $a^i = b^i(a^{-i})$ is "when the others' actions are a^{-i}, the best action which I can think of is a^i". The fact that a^{-i} is not in the domain of b^i means that the agent does not know how to respond to a^{-i} or alternatively will not regret her choice of action when the others played a^{-i}. Naturally we assume that each agent is capable of computing her own knowledge and henceforth that b^i is of degree d.

Definition 9 (feasible best response) *An action a^i for agent i is called* feasible best response *to a^{-i} if either a^{-i} is not in the domain of the agent's knowledge b^i or $u^i((b^i(a^{-i}), a^{-i})) \leq u^i(a)$.*

In other words, other actions may be better against a^{-i} but at least when choosing her action the agent was not aware of these.

The definition of feasibly dominant actions now follows naturally.

Definition 10 (feasibly dominant action) *An action a^i for agent i is called* feasibly dominant *if it is a feasible best response against any a^{-i}. We also call such an action FDA .*

It was argued in [13] that if an agent has feasibly dominant actions available, then it is *irrational* not to choose one of them.

4.5.2 When is it rational to tell the truth to the mechanism?

Recall that the overall utility of each agent i equals $g_{\hat{s}}(\tilde{v})$ where $\hat{s}$ denotes the chosen allocation and $\tilde{v} = (v^i, w^{-i})$. It is not difficult to see that when the agent declares a falsified valuation, there are cases where she will consequently lose. The agent needs therefore a good reason for lying to the mechanism. When the appeals of the agents are time-limited (i.e. of degree d) it was shown in [13] that the existence of FDAs for the agents *must* rely on further assumptions on the agents' *knowledge*. Here we formulate two such assumptions and show how to construct computationally efficient truthful FDAs for the agents.

Definition 11 *([13])* **(declaration based knowledge)** *Knowledge* $b^i(.)$ *is called* declaration based *if it is of the form* $b^i(w^{-i}, d^{-i}) = (w^i, d^i)$.

The semantics of declaration based knowledge is: "If I knew that the others declare (w^{-i}, d^{-i}), regardless of their appeals, I would like to declare (w^i, d^i)". In our OR bids example of section 3, such knowledge for Agent1 may be: "If Agent2 has a high valuation for item B, I would like to give it up".

A declaration based knowledge naturally defines an appeal function which we also denote by $b^i(.)$: $b^i(w, d) = (b^i(w^{-i}, d^{-i}), d^{-i})$.

Theorem 4.3 *If* $b^i(.)$ *is a declaration based knowledge for agent* i *then* (v^i, d^i, b^i) *is feasibly dominant for the agent.*

Proof: Let $\hat{s}$ denote the chosen allocation. Let $\tilde{v} = (v^i, w^{-i})$. Recall that the utility of agent i equals $g_{\hat{s}}(\tilde{v})$. Also let ϕ denote the empty appeal. Assume by contradiction that there exists $a^{-i} = (w^{-i}, d^{-i}, \phi^{-i})$ that contradicts the agent's knowledge. Note that the appeals of the other agents can be assumed empty and also that it must be that a^{-i} is in the domain of $b^i(.)$. Let $(w'^i, d'^i) = b^i(a^{-i})$. Let $s = k(d)$ and let $s' = k(d'^i, d^{-i})$ denote the allocation when she lies. By the assumption, $g_s(\tilde{v}) < g'_s(\tilde{v})$. However when the agent truthfully submits (v^i, d^i, b^i) the mechanism computes $s = k(d)$ and $s' = k(d'^i, d^{-i})$ and takes the better among them according to $\tilde{v}$. A contradiction. $\square$

Definition 12 *([13])***(appeal independent knowledge)** *Knowledge* $b^i(.)$ *is called* appeal independent *if it is of the form* $b^i(w^{-i}, d^{-i}) = (w'^i, d'^i, l^i)$.

Theorem 4.4 *If* $b^i(.)$ *is an appeal independent knowledge of agent* i *then there exists a truthful FDA of degree* d *for the agent.*

Proof: Define an appeal l^i as follows. Given (w^{-i}, d^{-i}) let $(w'^i, d'^i, l'^i) = b^i(w^{-i}, d^{-i})$. l^i computes $k(d'^i, d^{-i})$, $k(l'^i((w'^i, w^{-i}), (d'^i, d^{-i})))$ and takes the best according to (v^i, w^{-i}). Since all the functions involved are of degree d, so is $l^i(.)$. Similarly to theorem 4.3, (v^i, l^i) is an FDA. $\square$

The semantics of declaration based knowledge is the same as declaration based except that the agent also submits an appeal l^i.

Agents who are not capable of reasoning about others' appeals or do not want to count on them would have appeal independent knowledge. We argue that this would be the most common case. In all of the examples of section 4.4 the agents' knowledge was appeal independent.

5 Mechanism2: moving information around

A major difficulty that arises when coping with incomplete languages is the asymmetric knowledge of the agents regarding their *own* valuations. For example, in the setting of section 3, it is reasonable to assume that Agent1 can compute her own l_∞-approximation but Agent2 cannot compute it. Thus, Agent1 might face the following considerations:

- The result of the algorithm improves significantly when all agents report their l_∞-approximations.
- Reporting my l_∞-approximation instead of my truthful description, will enable Agent2 to compute the optimal result.

In other words, in mechanism1, agents may want to misreport in order to pass useful information about their own valuation to the others. In order to prevent this we modify the mechanism to allow the agents to convey such information.

Definition 13 (information structure) *An* information structure I^i *for agent i is a sequence of descriptions (possibly with repetitions)*

$(d_0, d_1, \ldots, d_k)$ *such that* d_0 *is a valid description.*

In addition we require each agent to provide for each d_j an *example* (w^{-i}, d^{-i}) such that $k(d_j, d^{-i})$ is a better allocation than $k(d_0, d^{-i})$. This is done in order to force the agents to submit only useful information.

I^i contains additional information that the agent can pass to the others' appeals. The semantics of I_i is "My valid description is d_0. Nevertheless, I suggest that you first try to work with d_1, after that with d_2, etc". Many alternative ways to define such information structures are possible. It may be interesting to compare between different structures.

We can now define our second mechanism.

Definition 14 (mechanism2) *Given an allocation algorithm* $k(d)$*, and consistency checker for the bidding language we define* mechanism2 *as follows:*

1. *Each agent submits to the mechanism:*
   - *an oracle* w^i*. (let* $w = (w^1, \ldots, w^n)$*)*
   - *an information structure* I^i*. (let* $I = (I^1, \ldots, I^n)$*)*
   - *an appeal function of the form* $l^i(w, I) = d$*.*
2. *The mechanism computes the allocations* $k(d), k(l^1(w, I)), \ldots, k(l^n(w, I))$ *and chooses among these allocations the one that maximizes the declared total welfare.*
3. *The mechanism calculates the payments according to the VCG formula.*

We can now expand the definition of knowledge under which the existence of truthful FDAs is guaranteed. This definition refers to knowledge that was obtained by checking a representative family of (tuples of) appeals of the other agents.

Definition 15 *[13]* **(d-obtainable knowledge)** *Knowledge $b^i(.)$ is called* d-obtainable *if the following holds:*

1. *b^i is of degree d.*

2. *Every appeal function that appears in the domain or in the range of $b^i(.)$, is of degree d.*

3. *There are at most n^d appeal functions that appear in the domain or in the range of $b^i(.)$. Moreover there exists a representative family L^i of no more than n^d $(n-1)$-tuples of appeals such that for every tuple φ^{-i} that appears in the domain of b^i there exists a $\psi^{-i} \in L^i$ such that for all (w^{-i}, I^{-i}), $b^i(((w^{-i}, I^{-i}), \varphi^{-i})) = b^i(((w^{-i}, I^{-i}), \psi^{-i}))$.*

The assumption that agents' knowledge is d-bounded is justified by the immense complexity of the appeal space. It assumes that an agent cannot think about more than a small family of representative cases L^i. For a more comprehensive discussion on this assumption see [13]. We need an additional assumption on the appeal class that the agent considers. We will remove this assumption later on.

Definition 16 (monotonic appeal) *We say that an appeal function $l(.)$ is* monotonic *if for every w and for every two structures $I = (I^1, \ldots, I^n)$ and $I' = (I'^1, \ldots, I'^n)$ such that I^j is a subset of I'_j for all j, $k(l(w, I'))$ is at least as good as $k(l(w, I'))$.*

In other words, giving more information to the appeal can just help it to compute a better result. We cannot expect the appeals to be monotonic as such monotonicity usually requires exponential time. However, it is reasonable to think that appeals will be monotonic *in general*, that is that the addition of useful information and, in particular, of truthful descriptions, usually helps the appeals to improve the overall result. Changing the order of the d_is in the information structures does not affect monotonic appeals.

Definition 17 (monotonic d-obtainable knowledge) *Knowledge for agent i is called* monotonic d-obtainable *if it is d-obtainable and all the appeals that appear in its domain or in its range are monotonic.*

Theorem 5.1 *If the agent's knowledge is monotonic d-obtainable, she has a truthful FDA of degree* $3 \cdot d$.

Proof: Let I^i denote a maximal sequence of useful information that an agent i can compute (i.e. it contains all the cases that the agent finds useful). Let b^i be a d-obtainable knowledge for agent i. Given (w^{-i}, I^{-i}) we shall define an appeal l^i as follows: Let L be the family of all appeals that appear in the domain or in the range of b^i. Let L^i be the representative family. We define ω to denote the set of all the "useful lies" $\omega = \{w^i | \exists \psi^{-i} \in L^i, \varphi^i s.t. (w^i, I^i, \varphi^i) = b^i(w^{-i}, I^{-i}, \psi^{-i})\}$. Obviously $|W|, |L|$ are bounded by a polynomial of degree d.

For every pair $(w^i \in \omega, l \in L)$ we let l^i compute the result of l as if she had submitted (w^i, I^i, l), i.e. compute $k(l(w^i, w^{-i}), (I^i, I^{-i}))$. The appeal returns the best of these allocations according to (v^i, w^{-i}).

As all the functions involved are of degree d, it is not difficult to verify that the appeal is of degree $3 \cdot d$.

We now show that submitting (v^i, I^i, L^i) is an FDA. Otherwise there exists a triplet (w^{-i}, I^{-i}, l^{-i}) that contradicts $b^i(.)$. Since $b^i(.)$ is d-obtainable we can assume that l^{-i} is in the representative family. Let $(w^i, \iota^i, \delta^i) = b^i(w^{-i}, I^{-i}, l^{-i})$. Because of the monotonicity we can assume that ι^i contains all the useful information that i can think of (i.e. $\iota^i = I^i$). However the appeal l^i *checks* the case where i submits (w^i, I^i, δ^i). Therefore l^is result must be at least as good as the result of the mechanism in this case – a contradiction. $\square$

6 Mechanism3: adding meta-appeals

In section 5 we assumed that the agents' appeals are monotonic. Our final step is to get rid of this assumption. We first define the notion of a meta-appeal.

Definition 18 (meta appeal) *A* meta appeal *is a function that gets a vector of information structures $I = (I^1, \ldots, I^n)$ and returns a list of vectors of the form $I' = (I'^1, \ldots, I'^n)$ such that I'^j is a subset of I^j.*

In other words, the meta appeals compute a list of alternative information structures for the group. Note that many variants of this definition are possible. We assume that all the meta-appeals are of degree d.

Definition 19 (mechanism3) *Given an allocation algorithm $k(d)$, and a consistency checker for the bidding language we define* mechanism3 *as follows:*

1. *Each agent submits to the mechanism:*
    - *An oracle w^i. (let $w = (w^1, \ldots, w^n)$)*
    - *An information structure I^i. (let $I = (I^1, \ldots, I^n)$)*
    - *An appeal function of the form $l^i(w, I)$.*
    - *A meta appeal $\chi^i(.)$.*
2. *The mechanism computes a list Γ containing all the results of the meta-appeals as well as the original tuple of information structures I.*
3. *The mechanism computes, for every pair (l^j, I') such that $I' \in \Gamma$ and l^j is an appeal, the allocation $k(l^j(w, I'))$. It also computes $k(d)$. It then chooses among these allocations the one that maximizes the declared total welfare.*

4. The mechanism calculates the payments according to the VCG formula.

Note that the mechanism is of degree $d+3$.

We can now define d-obtainable knowledge similarly to the previous section. We add however the condition that it ignores the meta appeals of the other agents.

Definition 20 *[13]* **(d-obtainable knowledge of mechanism3)** *We say that knowledge $b^i(.)$ of mechanism3 is d-*obtainable *if the following holds:*

1. *$b^i(.)$ is of degree d.*
2. *$b^i(.)$ ignores the meta-appeals of the other agents, i.e it is of the form $b^i(w^{-i}, I^{-i}, l^{-i}) = (w^i, I^i, l^i)$.*
3. *Every appeal function that appears, in the domain or in the range of $b^i(.)$, is of degree d.*
4. *There are at most n^d appeal functions that appear in the domain or in the range of $b^i(.)$. Moreover there exists a representative family L^i of no more than n^d $(n-1)$-tuples of appeals such that for every tuple φ^{-i} that appears in the domain of $b^i(.)$ there exists a $\psi^{-i} \in L^i$ such that for all (w^{-i}, I^{-i}), $b^i(((w^{-i}, I^{-i}), \varphi^{-i})) = b^i(((w^{-i}, I^{-i}), \psi^{-i}))$.*

The main justification behind the assumption that b^i ignores the meta-appeals is that the space of meta-appeals is extremely complex. Moreover, properties of the meta-appeals are only partially connected to the actual bidding language or the algorithm. The only potential profit from lying that we can imagine are "extra-trials" of the allocation algorithm when the others' appeals are forced to use the agent's false description. We presume that such potential gains are negligible compared to the obvious loss caused by lying. It is also natural to think that if the appeals of the other agents ignore the agent's recommendation to use d_1, they have a good reason to do so. We argue that knowledge which is not d-obtainable is unlikely to exist. However, this "thesis" needs to be checked *experimentally.*

Theorem 6.1 *If the agent's knowledge is d-obtainable, then she has a truthful FDA* (v^i, I, l^i, χ^i) *such that* l^i *is of degree* $3 \cdot d$ *and* χ^i *of degree* d*.*

Proof: Similarly to the proof of 5.1, given (w^{-i}, I^{-i}), we define the set of "useful lies" $\omega = \{w^i | \exists \psi^{-i} \in L^i, \varphi^i s.t. (w^i, I^i, \varphi^i) = b^i(w^{-i}, I^{-i}, \psi^{-i})\}$, and the family L of appeals which appear in $b^i(.)$. In addition we define the set of useful information structures $\chi^i = \{I^i | \exists \psi^{-i} \in L^i, \varphi^i s.t. (w^i, I^i, \varphi^i) = b^i(w^{-i}, I^{-i}, \psi^{-i})\}$. We define I to be a union of all $I \in \chi^i$, an appeal l^i like in the proof of 5.1. The proof that (v^i, I, l^i, χ^i) is an FDA is similar to 5.1.

□

6.1 Example: Mechanism3 with OR bids

Consider mechanism3 for CA with OR bids (section 3). Suppose that Agent1 notices the following phenomena:

1. When all agents perform l_∞-approximations the result of $k(.)$ usually improves considerably.
2. The result also typically improves if agents perform singleton concessions on different items. The improvement however is less significant than in the first case.

Such an agent may anticipate three kinds of appeals:

- Appeals of agents that notice the first phenomenon and will therefore leverage from her l_∞-approximation.
- Appeals of agents who notice only the second phenomenon and will only be disturbed by her l_∞-approximation.

- Appeals that will work best with her valid description.

Mechanism3 gives Agent1 the possibility of constructing a strategy that will **dominate** every case that she can think of! She just needs to include in her meta appeal three information structures. One that includes only her valid description, one that will include her singleton concessions as well and one that will also include her l_∞-approximation.

We note that there exist additional ways to justify why truth-telling is *the* rational strategy for the agents. Those are omitted from the paper mainly due to space constraints.

7 Other implementation issues

In this section we address two additional issues which a designer may face when implementing our mechanisms: guaranteeing individual rationality and forcing reasonable time limitations on the agents.

In [13] it was shown that the allocation algorithm can be transformed in polynomial time to an algorithm which satisfies additional monotonicity requirements. With such an algorithm it is possible to define the function $h^i(.)$ of our mechanisms similarly to Clarke's mechanism [1]. The proof that the resulting mechanisms satisfy individual rationality is similar to [13].

This paper shows that if enough computational time is given to the agents, they can construct truthful FDAs. On the other hand the mechanism needs to find a way to enforce *reasonable* time limits on the computational time of the agents, i.e. to enforce time limits on the appeals and meta appeals. This issue was discussed in [13]. In particular it was suggested that knowledge-reflecting structure will be chosen for description of the appeal functions. Such a structure enables the limitation of the computational time of the appeals according to the agents' own limitations and thus preserves the existence of truthful FDAs. We presume that severe limitations can be imposed on the length of the lists produced by the meta appeals while preserving the existence of FDAs. Finally, we think that it is a good heuristic to charge small fees for extra computational time.

At a first glance our protocols may seem to put a lot of burden on the agents. However we argue that with the right tools (e.g. tools for building oracles), mechanisms like ours can become even more "agent friendly" than non-revelation mechanisms.

8 Conclusions and further research

In this paper we propose a general way to overcome the deficiencies of VCG mechanisms with incomplete languages. Given an intermediate language, a consistency checker, and an algorithm for the computation of the outcomes (e.g. allocations) we construct three mechanisms, each more powerful but also more time-consuming than its predecessor. All our mechanisms have polynomial computational time and satisfy individual rationality.

We adopt the strong concept of feasible dominant strategies of [13] which is a bounded rationality version of dominant strategies and showed that under reasonable assumptions on the agents' knowledge, truth-telling is feasibly dominant for the agents. In addition when an agent lies to the mechanism, there are cases where she will consequently lose.

When the agents are truth-telling the results of our mechanisms are **at least as good** as the mechanisms' algorithm. Our methods are general and can be applied to any VCG , weighted VCG or compensation and bonus [14] mechanism.

The paper assumes that in practice, agents will have only limited knowledge and thus will not be able to do better than their truthful FDAs. This thesis can and should be checked by experiments with "real" agents. On the other hand we feel that this assumption will remain true even when severe time limitations are forced on the agents. In fact it will not even be a surprise if even in a VCG mechanism, if the bidding language and the allocation algorithm are reasonably designed, the agents will not be able to do better than truth-telling! This too can be checked experimentally.

Very little is currently known about the revenue of mechanisms for complex problems. In particular note that when a non-optimal VCG mechanism is naively used for a combinatorial auction, there are even cases where the mechanism must pay to the agents instead of vice-versa!

In our constructions, there are several tools that the designer must provide to the agents. Tools to construct oracles, descriptions, appeals etc. Methods for providing such tools were not discussed in this paper and are crucial for the success of our mechanisms.

Finally we note that it might be fruitful to explore the possibility of using appeal functions in situations where the agents have budget limits. When such limits exist, agents may have incentives to cause others to run out of budget and it is not likely that dominant strategy mechanisms exist. One natural way to deal with budget limits, is to truncate the agent's valuation to her limit and then use VCG [12]. Truth-telling in this mechanism is a *safe* strategy for the agent as she never pays more than her budget. We argue that appeals of certain forms can play the role of threats and prevent the worth-willingness of causing others to run out of budget.

Acknowledgments: We thank Yoav Shoham for many fruitful discussions. We thank Inbal Ronen for reviewing previous versions of this paper.

References

[1] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.

[2] Sven de Vries and Vohra Rakesh. Combinatorial auctions: A survey. To appear.

[3] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *IJCAI-99*, 1999.

[4] J. Green and J.J. Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, pages 427–438, 1977.

[5] T. Groves. Incentives in teams. *Econometrica*, pages 617–631, 1973.

[6] R. M. Harstad, Rothkopf M. H., and Pekec A. Computationally manageable combinatorial auctions. Technical Report 95-09, DIMACS, Rutgers university, 1995.

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

[8] A. Mas-Collel, Whinston W., and J. Green. *Microeconomic Theory*. Oxford university press, 1995.

[9] J. McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, pages 145–162, 1994.

[10] Paul Milgrom. Putting auction theory to work: the simultaneous ascending auction. Technical Report 98-002, Dept. of Economics, Stanford University, 1998.

[11] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proc. of the Second ACM Conference on Electronic Commerce (EC00)*, pages 1–12, 2000.

[12] Noam Nisan. Private communication, 2000.

[13] Noam Nisan and Amir Ronen. Computationally feasible vcg mechanisms. In *Proc. of the Second ACM Conference on Electronic Commerce (EC00)*, pages 242–252, October 2000.

[14] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behaviour*, 2001. To appear; Extended abstract appeared in STOC99.

[15] M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT press, 1994.

[16] David Parkes. ibundle: An efficient ascending price bundle auction. In *In Proc. ACM Conference on Electronic Commerce (EC-99)*, pages 148–157, 1999.

[17] Kevin Roberts. The characterization of implementable choise rules. In Jean-Jacques Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–349. North-Holland, 1979. Papers presented at the first European Summer Workshop of the Econometric Society.

[18] Amir Ronen. Mechanism design with incomplete languages. http://task.stanford.edu/, 2001. (full version).

[19] Toumas Sandholm. Issues in computational vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129, 2000.

[20] Tuomas W. Sandholm. Approaches to winner determination in combinatorial auctions. *Decision Support Systems*, to appear.

[21] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, pages 8–37, 1961.