Differential-Revelation VCG Mechanisms for Combinatorial Auctions

Wolfram Conen[1] and Tuomas Sandholm[2]*

[1] XONAR GmbH, Wodanstr. 7, 42555 Velbert, Germany, conen@xonar.biz
[2] Carnegie Mellon University, Computer Science Department,
5000 Forbes Avenue, Pittsburgh, PA 15213, sandholm@cs.cmu.edu

Abstract. Combinatorial auctions, where bidders can submit bids on *bundles* of items, are economically efficient mechanisms for selling items to bidders, and are attractive when the bidders' valuations on bundles exhibit *complementarity* and/or *substitutability*. Determining the winners in such auctions is a complex optimization problem that has received considerable research attention during the last 4 years. An equally important problem, which has only recently started to receive attention, is that of eliciting the bidders' preferences so that they do not have to bid on all combinations [6,8]. Preference elicitation has been shown to be remarkably effective in reducing revelation [13]. In this paper we introduce a new family of preference elicitation algorithms. The algorithms in this family do not rely on absolute bids, but rather on *relative (differential) value information*. This holds the promise to reduce the revelation of the bidders' valuations even further. We develop a differential-elicitation algorithm that finds the efficient allocation of items to the bidders, and as a side-effect, the Vickrey payments (which make truthful bidding incentive compatible). We also present two auction mechanisms that use differential elicitation: the *difference mechanism* and the *difference increment mechanism*.

1 Introduction

Combinatorial auctions, where bidders can submit bids on *bundles* of items, are economically efficient mechanisms for selling m items to bidders, and are attractive when the bidders' valuations on bundles exhibit *complementarity* (a bundle of items is worth more than the sum of its parts) and/or *substitutability* (a bundle is worth less than the sum of its parts). Determining the winners in such auctions is a complex optimization problem that has recently received considerable attention (e.g., [20,24,9,25,15,1,26]).

An equally important problem, which has received much less attention, is that of bidding. There are $2^m - 1$ bundles, and each bidder may need to bid on all of them to fully express its preferences. This can be undesirable for any of several reasons: (1a) determining one's valuation for any given bundle can be computationally intractable [21, 23,17,14]; (1b) there is a huge number of bundles to evaluate; (2) communicating the bids can incur prohibitive overhead (e.g., network traffic); and (3) bidders may prefer

* Dr. Sandholm's work was funded by, and conducted at, CombineNet, Inc., 311 S. Craig St., Pittsburgh, PA 15213.

J. Padget et al. (Eds.): AMEC 2002, LNAI 2531, pp. 34–51, 2002.

not to reveal all of their valuation information due to reasons of privacy or long-term competitiveness. Appropriate bidding languages [24,9,22,15,12] can potentially solve the communication overhead in some cases (when the bidder's utility function is compressible). However, they still require the bidders to completely determine and transmit their valuation functions and as such do not solve all the issues. So in practice, when the number of items for sale is even moderate, the bidders will not bid on all bundles.

We study the setting in which a benevolent auctioneer (or arbitrator) wants to implement an efficient allocation of a set of heterogeneous, indivisible goods. The preferences of the participating bidders (or consumers) are private information and utility is transferable via money. The auctioneer tries to design a mechanism that gives no incentive for the bidders to misreport preferences.

It is well known that the Vickrey-Clarke-Groves mechanism [27,5,10] (aka. Generalized Vickrey Auction (GVA)), *that is based on exhaustively eliciting all utilities*, is such an incentive compatible mechanism. However, in that mechanism, each bidder evaluates each of the exponentially many bundles, and communicates a value for each one.[1] This is clearly impractical even for auctions with moderate numbers of goods.

Consider the following: the (rational) preferences of bidders can be ranked (from most preferred towards least preferred). Each rank uniquely represents a bundle (bundles with consecutive ranks may have identical valuations). Combining the individual ranks will lead to combinations of ranks (respectively combinations of ranked bundles); some of them are feasible. All combinations form a lattice along a (weak) dominance relation. This lattice structure can be utilized to guide a (best-first) search through the space of (feasible and infeasible) combinations. This idea has been exploited in [6,8] to design an efficient, (individually) incentive compatible mechanism for combinatorial auctions. The mechanism may reduce the amount of elicited information in comparison to generalized Vickrey auctions. The mechanism asks each bidder for the (true) valuations of (a subset of) the bundles. We called this a *partial-revelation mechanism*. Recently, it has been shown that this method, and related elicitation methods, may lead to significant savings in the amount of information that is elicited from the bidders (compared to the full elicitation of the GVA)—in fact, because the number of items in the auction grows, only a vanishing fraction of all value queries end up being asked [13].

In this paper we present a mechanism that *does not elicit absolute valuations but rather elicits differences between valuations* and, thus, may reveal only a fraction of each value information to the auctioneer.[2] We call this *differential revelation* (because only differences of valuations are revealed). We present an algorithm to explore the rank lattice using differential value information. The algorithm determines an efficient allocation based on revealed valuation differentials. It also inherits the partial revelation properties of the algorithm discussed in [8], while saving the bidder from specifying absolute valuations. Note that in the worst-case, all valuation information has to be revealed—it is, however, a challenge to reduce this amount whenever possible. The algorithm was designed with this objective.

[1] In general, preference communication in combinatorial auctions is provably exponential (even to find an approximately optimal solution) in the theoretical worst case [16].

[2] This may be especially useful in settings where the communication between the bidder and the auctioneer is public.

We will also discuss the computation of Vickrey payments based on the information collected while executing the algorithm. We show that differential information suffices to determine the Vickrey payments and that all information necessary to compute the payments is already available once the algorithm has determined an efficient allocation. Before we proceed, an example will be given to demonstrate the basic ideas.

2 Example of *Differential* Elicitation

In this section we present an example of differential elicitation with three unit-demand bidders, three goods, and a benevolent auctioneer. Table 1 shows the valuations of the bidders for the goods and the individual rankings of the bundles.

Table 1. In this setting there are 3 unit-demanding consumers (allows us to neglect the valuations for the bundles) and 3 goods. The efficient allocation is to give good A to bidder 1, good B to bidder 2, and good C to bidder 3. Bidder-specific rankings for the bundles are given, for example the rank of good C is 1 for bidder 1 because good C is the good most preferred by bidder 1 (note that the ranking for bidder 1 is not unique, because the ranks of goods A and B could be swapped due to the indifference of bidder 1).

	A	B	C	∅
Bidder 1	**3**	3	4	0
Rank	3	2	1	4
Bidder 2	2	**5**	3	0
Rank	3	1	2	4
Bidder 3	3	4	**6**	0
Rank	3	2	1	4

With the valuations[3] and ranks shown in Table 1, a lattice of combinations of ranked preferences is given implicitly. The lattice is formed from the set of possible combinations of ranks and a (rank-)dominance relation that orders the combinations partially with respect to the ranks. A combination a *dominates* a combination b if and only if all the ranks in a are at least as low as the corresponding ranks in b. For example, $(2, 1, 1)$ (rank-)dominates $(3, 1, 1)$ and $(2, 1, 2)$, but not $(1, 3, 4)$. Note that if a dominates b, the value of the combination of bundles that is represented by a cannot be less than the value of the combination of bundles that is represented by b.[4] A search procedure is deployed that travels through the (implicitly given) lattice in a best-first manner, starting with the best possible combination and stopping with the first feasible combination found (combinations of ranks can represent feasible or infeasible combinations of bundles). In previous work, we have based the decisions which node to expand next on information about the total value of represented combinations. In the following, we will use information about the difference of the value of the considered combinations to the (undetermined) value of the best possible combination.

[3] We assume throughout the paper that valuations are integral.

[4] With the usual assumption of utility functions representing rational preferences.

With 3 bidders, the best possible combination of ranks is $(1, 1, 1)$. In the example, it represents the combination of the most preferred bundles (C, B, C).[5] This combination necessarily dominates all other combinations, both with respect to the ranks and the aggregated value (which is 15—this will not be revealed, however). Unfortunately, this combination is not feasible.

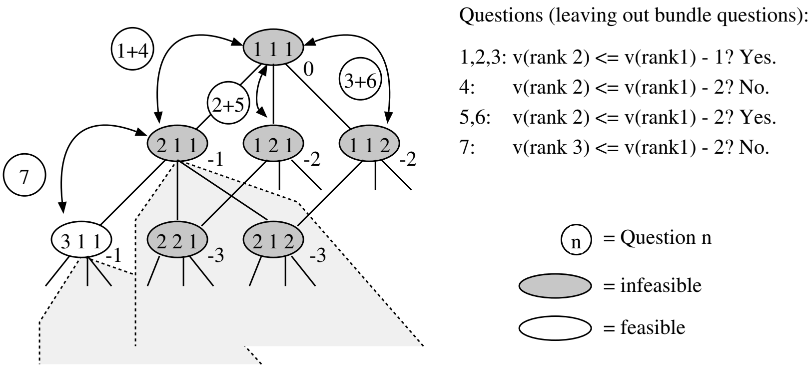

Fig. 1. Seven value queries are necessary to determine the efficient allocation in the above example (here and in the following, $v(rank\ r)$ denotes the valuation of the bundle at rank r). To the lower right of the combination nodes, the difference in value from the value of combination $(1, 1, 1)$ is shown. The combination with the smallest difference is the efficient allocation—here represented by the rank combination $(3, 1, 1)$. Note that as long as the nodes $(1, 2, 1)$ and $(1, 1, 2)$ aren't expanded (because other, at least equally promising nodes where chosen first), there is no need to expand the shaded areas either.

Now, consider the upper part of the rank lattice in Figure 1, which is used to guide the search for the efficient allocation. The auctioneer will ask queries of the following two types: (1) "*Give me the bundle at the currently considered rank*" (bundle query), and (2) "*Is the value of the currently considered bundle smaller than the value of your highest ranking bundle minus δ_i?*" (value query). We assume that it is common knowledge of the bidders and the auctioneer that the first bundle to be requested will be the bundle at rank 1 and that all δ_i are initially set to 1 and are incremented each time the bidder i answers "Yes".[6]

The search proceeds as follows. First, after asking the initial bundle queries and after discovering that the best combination is not feasible, all immediate successors of $(1, 1, 1)$ will be considered. These are the combinations $(2, 1, 1)$,$(1, 2, 1)$, and $(1, 1, 2)$. The currently considered rank is 2 for each bidder. The bidders will answer the bundle queries. Then the first round of value queries will be asked: bidder i, is your valuation of the bundle at rank 2 less or equal to your valuation of the bundle at rank 1 minus 1? Each

[5] Note that, in a strict sense, a bundle will be defined as a *subset* of the goods, and thus, one would have to write $(\{C\}, \{B\}, \{C\})$. We will relax this notational constraint whenever it can be done without harm (e.g., we will write AB instead of $\{A, B\}$, or A instead of $\{A\}$).

[6] Note that the queries given in the figure are more explicit than necessary. With the above conventions, the queries can be reduced to a signal "Next query" and the answer can be given as one bit (1=Yes, 0=No). Details will follow in Section 5.1.

bidder will answer with "Yes". Now the auctioneer knows that each immediate successor of $(1, 1, 1)$ represents a combination of bundles that has a value that is *at least* 1 less than the value of the best combination. To discriminate amongst the three successors, a second round of value queries is asked with all δ_i incremented to 2. This time, bidder 1 answers "No" (because the value of B, the bundle at rank 2, is only one less than the value of C, the bundle at rank 1). Bidder 2 and bidder 3 answer "Yes" again. This allows the auctioneer to fix the difference in value between $(1, 1, 1)$ and $(2, 1, 1)$ to 1 (we will also say that the δ of bidder 1 for the rank 2 is *tight*). In addition, it is now known that the difference in value between $(1, 1, 1)$ and the combinations $(1, 2, 1)$ and $(1, 1, 3)$ is *at least* 2. Because the search proceeds best-first, $(2, 1, 1)$ will be expanded next. Only two more queries are necessary: bidder 1 is asked for the bundle at rank 3 (the answer is A, which renders the considered combination $(3, 1, 1)$ feasible) and bidder 1 is asked again, if the value of the bundle represented by the currently considered rank (now rank 3) is 2 or more less than the value of the most preferred bundle. Again, the answer is "No". This allows to conclude that $(3, 1, 1)$ *must represent* an efficient allocation.[7] In total, the following information is revealed (to the auctioneer):

$$v_1(rank1) = v_1(rank2) + 1 = v_1(rank3)$$
$$v_2(rank1) \geq v_2(rank2) + 2$$
$$v_3(rank1) \geq v_3(rank2) + 2$$

This can also be summarized as follows:

	A	B	C	$\emptyset$
Bidder 1	$(x_1 - 1)$	$(x_1 - 1)$	x_1	0
Bidder 2	$\leq (x_2 - 2)$	x_2	$\leq (x_2 - 2)$	0
Bidder 3	$\leq (x_3 - 2)$	$\leq (x_3 - 2)$	x_3	0

The efficient allocation is determined to be $A \rightarrow 1$, $B \rightarrow 2$, $C \rightarrow 3$. Although the value of the allocation is not known, it can be bound to be at least 4 (because we have enough information to conclude that combination $(3, 2, 2)$ has a value that is at least 4 less than the value of $(3, 1, 1)$, and with the assumption of free disposal and values of 0 for empty bundles, the value of $(3, 2, 2)$ must be at least 0).

Interestingly, the Vickrey payments can be computed from the information already elicited. First consider bidder 1: because the allocated bundle B (respectively C) is the highest ranking bundle of bidder 2 (respectively 3), the payment of bidder 1 is 0. The effect of the participation of bidder 2 *and* 3 on bidder 1 is the known difference between $v_1(rank1)$ and $v_1(rank3)$, which is 1. Here, it is the participation of bidder 3 that causes this effect. This information is also available: the best possible feasible allocation if only bidder 1 and 2 participate is realized when both receive their highest ranking bundle (C respectively B). If bidder 3 enters the scene, bidder 2 will still receive this bundle while bidder 1 experiences a loss of 1 relative to realizing his most preferred alternative. The participation of bidder 2 has no such effect. The Vickrey payments are therefore $t_1 = 0, t_2 = 0$, and $t_3 = 1$.

---

[7] Note that the available information would also suffice to conclude that $(3, 1, 1)$ represents the only efficient allocation, because we know that the value of the combinations represented by $(2, 2, 1)$ and $(2, 1, 2)$ is at least 2 less than the value of the best combination.

In section 4, we will prove that the *differential* information collected by our algorithm (primarily used to determine the efficient solution) will *always* suffice for determining the Vickrey payments. Note that in the above example *no* absolute valuation is revealed.[8] Furthermore, queries that are directly related to the bundles at rank 3 of bidders 2 and 3 are never asked.

In the following section, we will first specify the underlying model more precisely and then present the basic algorithm. We consider Vickrey payments in section 4. In section 5, we discuss how the algorithm can be used to create mechanisms that use different type of queries to elicit differential value information from the bidders. In section 6, properties of the mechanism related to incentive-compatibility, information revelation, communication complexity and the cognitive burden of bidders are discussed. Section 7 relates our research to ascending combinatorial auctions and section 8 offers some conclusions and a brief outlook.

3 The Model

Our setting is based on the concepts introduced in [6]. We consider an economy consisting of n consumers, $N = \{1, \dots, n\}$, a seller 0, and m goods, $\Omega = \{1, \dots .m\}$. Each consumer $i \in N$ has utility over bundles, given by a function $v_i : 2^\Omega \to \mathbb{N}_0^+$, where $v_i(\emptyset) = 0$.[9] The seller has all the goods; the consumers own none of them. We will neglect the seller. If the seller has reservation values for bundles, he can be modeled as an additional consumer.

It is well known that a bidder's preferences can be represented by a utility function only if the preference order is *rational*, that is, the preference order over alternatives is transitive and defined on all pairs of alternatives (equal preference between alternatives is fine). This preference order induces a rank function as follows.

Definition 1 (Rank function, Inverted Rank function). *Let R be the set of the first 2^m natural numbers, $\{1, \dots, 2^m\}$. Let, for every bidder i, the rational preference order $\succsim_i$ be defined over 2^Ω. A bijective function $r_i : 2^\Omega \to R$ will be called the* rank function *for bidder i if it assigns a unique number (*rank*) to each bundle, such that, for every pair $x, y \subseteq \Omega$ of bundles with $x \succ_i y$, $r_i(x) < r_i(y)$ holds. The inverse r_i^{-1} of r_i gives the bundle that corresponds to a rank.*

A rank function and its inverse exist for every rational preference relation. In the presence of indifference, there is no unique rank function. A combination of ranks of the bidders can be viewed as representing a potential solution to the allocation problem at hand. Some of these potential solutions are invalid, though. The others determine *allocations*.

[8] Note that an auction with a single item only is a degenerated case with only one difference per bidder and is, therefore, not a good example. Because at least one difference of every bidder will be a target of elicitation queries, no difference can be left untouched (so, the revelation is not partial). The absolute valuations of all loosing bidders will be revealed. Revelation of the valuation of the winning bidder can be limited to a fraction of its value, which suffices to beat the second largest difference (so, the revelation can be fractional).

[9] We will later add assumptions of quasi-linearity and free disposal to this setting.

Definition 2 (Combination of Ranks). *Let C be the set of all possible n-ary tuples over R, that is $C = R \times \cdots \times R$. An element $c \in C$, $c = (r_1, \ldots, r_n)$ will be called* combination of ranks. *(If, for every position i of c, the corresponding function r_i^{-1} is applied, a* collection of bundles, *b^c, will be obtained.)*

Definition 3 (Feasible Combination). *A combination c of ranks is* feasible *if the corresponding collection b^c of bundles is a partition (possibly with empty elements) of a (not necessarily proper) subset of Ω. A feasible combination determines an* allocation *X^c with $X_i = b^c[i]$, $i = 1, \ldots, n$, where the seller keeps the item set $X_0 = \Omega / \bigcup_i b^c[i]$. Here, $[i]$ denotes the i'th element of the tuple.*

Definition 4 (Dominance of Rank Combinations). *A binary relation $\succeq \subseteq C \times C$ will be called a* dominance relation *if, for all $x, y \in C$, $(x, y) \in \succeq$ if and only if $x[i] <= y[i]$ for all $i \in N$.*

The relation $\succeq$ is a *partial order* and C forms a *complete lattice* with respect to $\succeq$ with $lub(C) = (2^m, \ldots, 2^m)$ and $glb(C) = (1, \ldots, 1)$.

The observation exploited by the following algorithm is that a combination that is feasible and not dominated by any other feasible combination is *Pareto-efficient*. In addition, if utilities are considered, the welfare-maximizing allocation will be among these feasible Pareto-efficient combinations. The following algorithm will search the space of combinations (including infeasible ones) to determine a welfare-maximizing (or efficient) allocation.

3.1 Computing an Efficient Allocation

We assume a setting with transferable utility (the utility functions are quasi-linear in money). The algorithm to be presented is essentially a modification of the **EBF** algorithm that we studied in [8], and inherits some of its properties. For presentation purposes, the **EBF** algorithm is repeated here and the necessary modification is outlined afterwards.

Algorithm **EBF (Efficient Best First)**:
(1) OPEN $= \{(1, \ldots, 1)\}$;
(2) **loop**
(3) **if** $|$OPEN$| = 1$ **then** $c =$ combination in OPEN
    else Determine
      $M = \{k \in OPEN | v(k) = \max_{d \in OPEN} v(d)\}$. /* to be modified below */
      if $|M| \geq 1 \wedge \exists d \in M$ with *Feasible(d)* **then return** d
      else Choose $c \in M$ such that no $d \in M$ exists with $d \succ c$;
    OPEN $=$ OPEN $\setminus \{c\}$.
(4) **if** *Feasible(c)* **then return** c
(5) SUC $= \text{suc}(c)$
(6) **foreach** $n \in$ SUC **do**
      if $n \notin$ OPEN **then** OPEN $=$ OPEN $\cup \{n\}$

We recently showed that **EBF** algorithms determine an efficient (that is, welfare-maximizing) allocation [8]. Note that we speak of algorithm*s* because the choice of the next combination to be expanded is not deterministically specified above if 2 or more combinations have the same value. So, actually a family of algorithms, differing with respect to their tie-breaking rule, is given above.

The following observation motivates the changes to the algorithm that will be outlined below.

Observation: *The determined (feasible) collection does not only represent an allocation that maximizes aggregated utility, it also minimizes (over all feasible collections) the aggregated loss in utility of the bidders (here, the individual loss of a bidder is the difference between the utility of the bidder's most preferred bundle and the utility of the bundle the bidder receives in the considered allocation).*

We will now modify step 3 of the algorithm such that the algorithm can make use of differences in utility when deciding which combination to explore next (in contrast to [8], where it has been assumed that all precise, absolute value information that is necessary to determine the subset of the most efficient combinations from the set of currently considered combinations, that is OPEN, is elicited).

Assume for simplicity that a set of quadruples is used to collect the elicited information. Each quadruple (i, b, δ, s) identifies a bidder i, a bundle b, a value for the information about the difference between the valuation of i for the highest ranking bundle and i's valuation for bundle b, and a bit s that shows if δ is the precise difference (s=1) or a lower bound (s=0). There are quadruples for every bidder-bundle pair (we will also write δ_i^b and s_i^b if we want to access the information in the quadruple for a given (bidder,bundle)-pair (i, b)).

Initially, δ_i^b and s_i^b will be set to 0 for all $i \in N, b \subseteq \Omega$. To simplify the presentation, we will assume that all rank functions and their inverses are known (i.e., we can determine a bundle from a rank information, this allows us to neglect rank/bundle queries for now). For each i, the bit s of the highest ranking bundle $r_i^{-1}(1)$ will be set to 1. We assume that the information in the quadruples is and remains consistent, that is, the following condition invariantly holds:

$$\delta_i^b = v_i(r_i^{-1}(1)) - v_i(b), \text{ if } s = 1 \text{ and } \delta_i^b \leq v_i(r_i^{-1}(1)) - v_i(b), \text{ if } s = 0$$

Now, assume that the combination $(1, \ldots, 1)$ is not feasible. The algorithm will now add all successors of this combination to the set OPEN. Each successing combination will cover one (bidder,bundle)-pair for which $s = 0$ holds (one such pair per bidder).

Note that the value of each combination $c = (c_1, \ldots, c_n)$ can be related to the (unknown) value of $(1, \ldots, 1)$ by computing the sum of the deltas in the quadruples that are covered by c. This set of covered quadruples is

$$Q^c = \{(i, b, \delta, s) \mid i \in N, b = r_i^{-1}(c_i)\}.$$

The sum of difference Δ^c can be determined as $\sum_i \delta_i^{r_i^{-1}(c_i)}$ (i.e., the sum of the δs in Q^c). We know (assuming the information has been truthfully reported) that the value of c is at least Δ^c currency units lower than the value of $(1, \ldots, 1)$. If all s in Q^c are 1, the difference is tight, that is $v(c) = v((1, \ldots, 1)) - \Delta^c$, otherwise $v(c) \leq v((1, \ldots, 1)) - \Delta^c$ (canonically extending the value function to combinations of ranks by setting $v(c) = \sum_i v(r_i^{-1}(c_i))$).

Consider again the set OPEN. In step 3, a set M will be determined that contains all elements from OPEN such that the difference is tight and has the same value for each combination in M and every other combination in OPEN has a difference Δ (tight or otherwise) that is higher in value.

The modification of step 3 is as follows (the set M is completely determined):

(3 – old version)
$M = \{k \in OPEN | v(k) = \max_{d \in OPEN} v(d)\}$.

is changed to

(3 – new version)
$M = \{k \in OPEN|$ *Tight*[10]$(k) \wedge \Delta^k \leq \Delta^d$ for all d with *Tight*(d)
$\wedge\ \Delta^k < \Delta^d$ for all d with *Not*(*Tight*(d)) $\}$.

The information necessary to determine the set M (which is always non-empty) will be collected by a mechanism which builds upon the modified algorithm. Such mechanisms will be outlined below. Assume now that we have a method that allows us to correctly determine the set M in each iteration of the algorithm. This allows us to show that the following result holds in the context of transferable utility:

Proposition 1. *Any algorithm of the modified* **EBF** *family (to be called* **EBF-DE** *where DE stands for* differential elicitation*) determines an efficient (that is, welfare-maximizing) allocation.*

This follows, using Proposition 8 from [8] (which states that any EBF algorithm determines an efficient allocation), from the fact that the set M in the original version and in the new modified version coincide in each iteration. To see this, the following proposition is helpful:

Proposition 2. *Given an arbitrary but fixed subset S of the set of all possible combinations C. Then the set M, if determined from S (substitute S for OPEN above) with the old version, coincides with the set M determined with the new version.*

Proof. Let M^{old} be the set determined from S by the old version. Let c^* be one of the combinations in C with the highest value $v^* = v(c^*)$. (a) For each $m \in M^{old}$, $v^* - v(m) \leq v^* - v(n)$ holds for all $n \in S$. Assume that a protocol is available to elicit information from the bidders and whose use guarantees that the following invariantly holds: (b) $\Delta^k \leq v^* - v(k)$ for all $k \in S$ and (c) *Tight(k)* = *TRUE* if and only if $\Delta^k = v^* - v(k)$ for all $k \in S$. Assume that the protocol has been used to determine a set M^{new}. To show the proposition, first assume that a combination $c \in M^{old}$ exists such that $c \notin M^{new}$. (d) From the conditions for the formation of M^{new} follows that a combination $k \in S$ exists with a tight[11] Δ^k such that $\Delta^k < \Delta^c$. From (a),(b) and (c) follows $\Delta^c \leq v^* - v(c) \leq v^* - v(k) = \Delta^k$, contradicting (d) respectively the assumption. The other case is equally straightforward. □

[10] *Tight*(k) answers TRUE, if all s in the quadruples covered by k are 1, see above.

[11] Such a k with a tight Δ exists because for any $d \in S$ with $\Delta^d < \Delta^c$ and Δ^d is not tight, another d' exists with $\Delta^{d'} < \Delta^d$. With the finiteness of C and the conditions for M, at least the Δ of one such d' must be tight. This is the k used above.

With the fact that each set OPEN that can occur during the execution of the algorithm is necessarily a subset of C, the above Proposition 1 follows.

We will later discuss how the algorithm can be embedded into a mechanism that computes an efficient allocation and elicits differential valuation information only. But first, we will show that the *(differential) information collected while executing an* **EBF-DE** *algorithm is sufficient to determine Vickrey payments.*

4 Determining the Vickrey Payments Based on Differential Information Only

Now we will turn our attention to the information that is required to determine Vickrey payments. Recall that the Vickrey payment of bidder i reflects the effect of her participation in an economy E: a consumer i will pay an amount equal to the utility that the other consumers will loose due to the participation of i, that is

$$t(i) \;=\; V(E_{-i}) \;-\; \sum_{j \in N, j \neq i} v_j(X_j) \tag{1}$$

where E_{-i} is the economy E without i and $V(E_{-i})$ is the utility that can be realized implementing a welfare-maximizing allocation for E_{-i}.

We will now assume that an execution of an **EBF-DE** algorithm has determined an efficient allocation X for an economy E.

Proposition 3. *No information in addition to the information already obtained by* **EBF-DE** *is necessary to determine the Vickrey payments.*

Proof. We assume that c is the solution combination that was found by the algorithm and that it represents the allocation X. The tight difference between the value of the combination of the highest ranking bundles and the value of c is known to be Δ^c. First, note that *tight differential value information for all combinations with higher value (resp. smaller Δ) than c have already been obtained* (a consequence of the efficiency of EBF-DE algorithms. Below, we will refer to the italicized claim as (a)).

Assume that consumer i is removed from economy E to form E_{-i} and that c^{-i} refers to a combination of ranks that represents an efficient allocation $X^{-i} = (X_1^{-i}, \ldots, X_{i-1}^{-i}, X_{i+1}^{-i}, \ldots, X_n^{-i})$ for E_{-i}.

First, observe that (1) can be re-written as follows:

$$\begin{aligned}
t(i) &= \sum_{j \in N \setminus \{i\}} v_i(X_j^{-i}) \;-\; \sum_{j \in N \setminus \{i\}} v_j(X_j) \\
&= \sum_{j \in N \setminus \{i\}} \left[v_i(r^{-1}(1)) - \left(v_i(r^{-1}(1)) - v_i(X_j^{-i})\right)\right] \;- \\
&\qquad \sum_{j \in N \setminus \{i\}} \left[v_i(r^{-1}(1)) - \left(v_i(r^{-1}(1)) - v_j(X_j)\right)\right] \\
&= \sum_{j \in N \setminus \{i\}} v_i(r^{-1}(1)) - \sum_{j \in N \setminus \{i\}} \left(v_i(r^{-1}(1)) - v_i(X_j^{-i})\right) \;- \\
&\qquad \sum_{j \in N \setminus \{i\}} v_i(r^{-1}(1)) + \sum_{j \in N \setminus \{i\}} \left(v_i(r^{-1}(1)) - v_j(X_j)\right) \\
&= - \sum_{j \in N \setminus \{i\}} \left(v_i(r^{-1}(1)) - v_i(X_j^{-i})\right) + \sum_{j \in N \setminus \{i\}} \left(v_i(r^{-1}(1)) - v_j(X_j)\right) \\
&= \sum_{j \in N \setminus \{i\}} \left(v_i(r^{-1}(1)) - v_j(X_j)\right) - \sum_{j \in N \setminus \{i\}} \left(v_i(r^{-1}(1)) - v_i(X_j^{-i})\right) \\
&= \sum_{j \in N \setminus \{i\}} \delta_j^{X_j} - \sum_{j \in N \setminus \{i\}} \delta_j^{X_j^{-i}} \quad \text{(setting } \delta_j^A = v_j(r^{-1}(1)) - v_j(A)\text{)} \\
&= \sum_{j \in N, j \neq i} \delta_j^{r_j^{-1}(c_j)} - \sum_{j \in N, j \neq i} \delta_j^{r_j^{-1}(c_j^{-i})} \quad (c \text{ represents } X,\ c^{-i} \text{ represents } X^{-i})
\end{aligned}$$

The first term does not require additional information: it is the sum of the tight δs of the bidders other than i for the combination c. Further assume that a reduced $(n-1)$-ary allocation $Y^{-i} = (Y_1, \ldots, Y_n)$ (leaving out the bidder i respectively its index) can be found with a tight difference that is smaller than $\sum_{j \in N, j \neq i} \delta_j^{r_j^{-1}(c_j)}$. Further assume that additional differential value information would be required to find this reduced allocation. Then a combination $d = (Y_1, \ldots, X_i, \ldots, Y_n)$ could be constructed that would have a smaller difference (resp. a higher value) than X and that would have required additional information to establish its difference, thus contradicting (a). □

5 Auction Mechanisms That Elicit Valuation *Differentials*

We have an algorithm available to determine efficient allocations and demonstrated how the collected information can be used to compute Vickrey payments. To further complete the ingredients necessary for the design of a mechanism, we briefly outline two elicitation mechanisms that can be used to collect the information that is required in step (3) of the algorithm. The presentation uses the quadruple data structure suggested above.

5.1 Difference Increment Mechanism

In the algorithm, in each step (3), the complete set of combinations with the currently smallest tight difference is determined. It is also possible to determine (in each round) one element of M only, which can be done by finding one combination c such that the bound is tight and every other bound has the same or a lower value (without considering tightness). In both cases, non-tight differences (i.e., bounds of differences) need to be tightened. To tighten (or to lower) a bound for a combination c covering a non-tight quadruple $(i, r_i^{-1}(c_i), \delta, 0)$, the following query will be submitted to bidder i:

Is the difference between your valuation of the highest ranking bundle and the valuation for the bundle at rank c_i larger than δ?

If the answer is "yes", δ can be incremented. If the answer is "no", s will be set to 1 (s corresponds to the last position in the quadruple and indicates whether the difference is tight ($s = 1$) or a lower bound ($s = 0$)).[12]

5.2 Difference Mechanism

Instead of incrementing δ stepwise, the precise difference can be requested (each s can be set to 1 permanently):

Give me the difference between your valuation of the highest ranking bundle and the valuation for the bundle at rank c_i?

6 Properties of Our Differential Elicitation Mechanisms

In this section we discuss the correctness, amount of information revealed, and the communication complexity of the above two differential elicitation mechanisms. We also address the cognitive burden imposed on the bidders.

6.1 Efficiency and Incentive Compatibility

Both types of queries can be utilized in step (3) of **EBF-DE** to establish an elicitation protocol (resulting from an elicitation policy, see below) that satisfies the conditions (b) and (c) that were used in the proof of the efficiency of **EBF-DE** (Proposition 2). A straightforward policy to determine the next bidder to be queried is to pick one of the collections with the smallest, non-tight Δ and, then, pick one of the covered (bidder,bundle)-pairs with non-tight δ. This has to repeated until enough information is available to safely determine the set M.[13]

Assuming that enough space is available to store the elicited information (so that queries do not have to be repeated), the underlying **EBF-DE** algorithm induces a natural sequence of queries, starting with queries related to the most preferred bundles and proceeding step by step to the least preferred bundles. Note that the details of the selection

[12] Note that this query can be transmitted in one bit, see the discussion in Section 6.3.

[13] Because the tightness of the considered δ is established immediately (in the difference mechanism) or eventually (in the difference increment mechanism) and because the bound is incremented without leaving ranks (respective bundles) and their valuation unconsidered, the finiteness of the information suffices to establish that the policy, together with one of the query types, determines an elicitation protocol that satisfies the above mentioned conditions.

of the next bidder to be asked are left out here and should be specified together with the chosen tie-breaking rule, when the mechanism is implemented. For the properties that are of interest in the context of this paper, namely efficiency and incentive compatibility, discussing these details is not necessary, as the following proposition demonstrates (an immediate consequence of the preceding propositions):

Proposition 4.
RANK-DE[14] *mechanisms are incentive compatible and economically efficient.*

6.2 Amount of Information Revealed

The difference increment mechanism generally reveals less (and never more) information to the auctioneer than the difference mechanism. Obviously the difference increment mechanism will not reveal more because the same information is elicited incrementally rather than at once. The difference increment mechanism may reveal strictly less because incrementing the bound but still not meeting the tight difference may already exclude a bundle from being in the efficient allocation—in which case no more information about that difference is elicited.

6.3 Communication Complexity

The difference increment mechanism can be implemented by communicating one bit at a time between the auctioneer and a bidder. Assume that the bidders and the auctioneer adopt the following conventions: once the mechanism is initiated, only single bits will be transmitted from the auctioneer to the bidders. The bit will signal to the bidder that an answer to the following query is expected: *Is the difference in value between your highest ranking bundle and the bundle at the rank that is currently considered, larger than the currently considered difference?* All participants (that is, the bidders and the auctioneer) will first set the difference to 0 and the considered rank to rank 2. If the bidder answers with "Yes", the difference is incremented. If the bidder answers with "No", the next less preferred rank will be considered for the next query. In addition, the auctioneer now knows (from the properties of the rank function and the above conventions) that the currently considered difference is tight for the currently considered rank (which will be incremented consequentially). In the worst case, a bidder will have to send $2^m - 1$ "No"s (=0). The maximum number of "Yes" answers depends on the valuation for the highest ranking bundle, say v. In the worst case, v "Yes" bits will be transmitted.

In the difference mechanism, an agent will receive at most $2^m - 1$ queries, and each of these can be answered with a number in the interval $[0, \ldots, v]$. The sum of the numbers cannot exceed v and will equal v in the worst case. In binary encoding, the numbers will be transmitted with logarithmic size, bounded from below by $\lceil \log_2 v \rceil + (2^m - 2)$ (maximal difference v for the bundle at rank 2; 0 for all other bundles, $m > 1$) and bounded from above by $(2^m - 1)\lceil \log_2 \lceil \frac{v}{2^m - 1} \rceil \rceil$.

[14] We call mechanisms that consciously explore the rank lattice **RANK** mechanisms. The specific family of mechanisms presented here is based on the family of **EBF-DE** algorithms. The postfix DE refers to the *differential elicitation* of value information that is utilized by the mechanisms.

To compare the query types, consider an agent who receives differential increment queries until a tight difference for a certain bundle is established. Then it is generally true that establishing this tight difference would have required less communication if difference queries would have been used. This can be seen by substituting the established difference for v and the rank of the bundle for 2^m in the above formulas. Intuitively, the "No" answers in the difference increment mechanism establish an unary encoding of the difference and the "Yes" answers count the higher-ranking bundles one by one, while the answers to the difference questions form a set of binary-encoded numbers that add up to the difference. The cardinality of the set is the rank of the bundle minus 1.

On the other hand, if the problem allows to stop querying difference increment questions with some non-tight bound, querying (tight) differences directly may be more costly. A straightforward example can be constructed with a single item and two agents. Let the utility of the item for agent 1 be 1000, and 2 for agent 2. A difference mechanism would elicit the answers 1000 and 2 (13 Bits). A difference increment mechanism would require and receive two "Yes" and one "No" answers from agent 2 and three "Yes" answers from agent 1 (6 Bits) to establish an outcome.

6.4 Cognitive Burden of the Bidders

Partial preference elicitation reduces the cognitive burden of the bidders because they do not have to evaluate all bundles. Furthermore, for some bundles that have to be evaluated, fractions of the valuations suffice. This is in sharp contrast to direct-revelation mechanisms such as the GVA, where all bundles have to be evaluated exactly. The elicitation method introduced in this paper further improves over previous elicitation methods [6,8,13] in that only valuation *differences* need to be determined (fractionally).

An additional note regarding the cognitive burden of the bidders seems in order. The mechanism does not require *per se* that the bidders are aware of the complete ranking of their preferences. In fact, a bidder may start with a rather rough categorization of his preferences (say: this is my most preferred choice, these 3 choices are really nice, these 4 choices are not bad, the rest is not very attractive) that may need to be refined during the course of the mechanism.[15] In the worst case, a bidder may need to reveal her full rank and utility functions. As related experiments have demonstrated (see [13]) this seems to not be necessary very often. In the best case, only the bundle at rank 1 of

[15] It would be nice if this refinement process would be consistent with the complete preference relation of the bidder (that is: no bundle is ranked erroneously). However, because inconsistence would only be observable if monotony is violated, this is not detectable generally. Considering observability, this is not really relevant – the mechanism will still compute an allocation that is efficient with respect to the reported preferences (though, from an idealistic point of view, efficiency might not be achieved due to erratic behavior of a bidder). In a less static context, any preference relation may change during the execution of a coordination mechanism. On the positive side, the mechanism presented here is robust against such changes in the sense that no re-computation will be performed and no query that has been asked already will be repeated. On the negative side, a bidder with changed preferences can not influence the partial/differential information that has been elicited already. She can, however, adapt her future answers to the new situation (of course, this may break efficiency).

each bidder (not even the valuation) will be revealed (though this case does not seem to be very likely as well). In the other cases, rank and utility function will only be partially revealed and, as has been argued above, will not necessarily be fully determined by the bidders.

7 Related Research: Ascending Combinatorial Auctions

An important related research thread tries to identify iterative or progressive auction protocols that try to limit the space of preferences that are to be revealed in comparison to the fully revealing, naive GVA. Recently, auctions that follow certain solution procedures for primal/dual linear programs have been studied extensively and with respect to incentive compatibility (see [4] for an overview and new suggestions, or consider [2]). Another approach (AkBA) has been suggested [28], though a detailed analysis of incentive compatibility properties has yet to be performed. Iterative auctions are price-based, which requires that to guarantee that Vickrey payments are determined by the auction, prices must be computable that coincide with the Vickrey payments (incentive-compatibility!). Depending on the allowed price structures[16], equilibrium prices (that solve an underlying dual model) may or may not exist. The existence depends on properties of preferences which will be considered either individually (e.g. "gross substitutes") or, more general, with respect to the combination of bidder types ("agents are substitutes", [3]). The allowed price structure will also influence the applicability of the suggested mechanisms. For example, the unconstrained anonymous prices used in AkBA, see [28], may require an enforcement of the condition that each bidder is only allowed to purchase one bundle in one transaction (at the price quoted for that bundle and not, for example, in two transactions as may seem attractive if the sum of prices for sub-bundles is below the quoted price). Similar considerations are necessary for the unconstrained non-linear (and non-anonymous) prices used in auctions based on and significantly extending the work of Bikhchandani et. al (see, for example, [3,18,19,4]). The immediate computation of Vickrey payments, as it is enabled by our mechanisms, renders such considerations unnecessary.

8 Conclusions and Future Research

Combinatorial auctions, where bidders can submit bids on *bundles* of items, are economically efficient mechanisms for selling items to bidders, and are attractive when the bidders' valuations on bundles exhibit *complementarity* and/or *substitutability*.

[16] Possible price structures include the following cases: (1) the price for a bundle is additive in the prices of the contained goods [11]; (2) unconstrained, non-linear prices for every bundle [28] are given, only one bundle may be purchased per agent; (3) coherent prices for bundles are given where the price for a bundle may not exceed the sum of prices of the bundles in any partition of the bundle, and where the prices for super bundles of the bundles in the supported allocation are additive [7]. Furthermore, prices can be differentiated to a different degree, including (a) anonymous prices; (b) different prices for the set of buyers and the set of sellers; or (c) different prices for each individual agent.

An important problem, which has only recently started to receive attention, is that of eliciting the bidders' preferences so that they do not have to bid on all combinations. Preference elicitation has been shown to be remarkably effective in reducing revelation.

In this paper we introduced a new family of preference elicitation algorithms, which utilizes the general structure of the previously studied partial-revelation mechanism and inherit some of their properties.[17] However, the algorithms in the family presented here do not rely on absolute bids, but rather on *relative (differential) value information*. This holds the promise to reduce the revelation of bidders' evaluations even further. In addition, it may be easier for the bidder to determine differences in valuation than to determine the absolute level of valuations.[18] We developed a differential-elicitation algorithm that finds the efficient allocation of items to the bidders, and as a side-effect, the Vickrey payments (which make truthful bidding incentive compatible). Finally, we briefly presented two auction mechanisms that use differential elicitation: the *difference increment mechanism* and the *difference mechanism*.

Future research involves studying this new elicitation format experimentally and theoretically to determine how much revelation it saves in practice on real-world problems. It also includes developing additional query types and, if possible, yet more effective elicitation policies. Future research also includes developing a deeper understanding of how preference elicitation and ascending auctions are related, and if possible, developing effective hybrid auctions that use both techniques.

References

1. Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 39–46, Boston, MA, 2000.
2. Lawrence M Ausubel and Paul Milgrom. Ascending auction with package bidding. Technical report, University of Maryland, Department of Economics, 2002. Version April 25.
3. S. Bikhchandani and J.M. Ostroy. The package assignment model. UCLA Working Paper Series, mimeo, 2001.
4. Sushil Bikhchandani, Sven de Vries, James Schummer, and Rakesh V. Vohra. Linear programming and Vickrey auctions, 2001. Draft.
5. E H Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
6. Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 256–259, Tampa, FL, October 2001. A more detailed description of the algorithmic aspects appeared in the IJCAI-2001 Workshop on Economic Agents, Models, and Mechanisms, pp. 71–80.
7. Wolfram Conen and Tuomas Sandholm. Coherent pricing of efficient allocations in combinatorial economies. In *AAAI-02 workshop on Game-Theoretic and Decision-Theoretic Agents*, Edmonton, Canada, 2002.
8. Wolfram Conen and Tuomas Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, 2002.

[17] Compare [8] for results related to the efficiency of elicitation.

[18] However, if a bidder had to determine *all* differences, the two cases require identical effort.

9. Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 548–553, Stockholm, Sweden, August 1999.
10. Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
11. F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87:95–124, 1999.
12. Holger Hoos and Craig Boutilier. Bidding languages for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1211–1217, Seattle, WA, 2001.
13. Benoit Hudson and Tuomas Sandholm. *Agent-Mediated Electronic Commerce IV*, chapter Effectiveness of Preference Elicitation in Combinatorial Auctions. Number 2531 in Lecture Notes in Computer Science (LNCS). Springer, 2002. Available in this volume. Extended version: Carnegie Mellon University, Computer Science Department, CMU-CS-02-124, March. Also: Stanford Institute for Theoretical Economics workshop (SITE-02).
14. Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 169–182, Sienna, Italy, July 2001.
15. Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 1–12, Minneapolis, MN, 2000.
16. Noam Nisan and Ilya Segal. The communication complexity of efficient allocation problems, 2002. Draft. Second version March 5th.
17. David C Parkes. Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce Workshop at the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.
18. David C Parkes and Lyle Ungar. Iterative combinatorial auctions: Theory and practice. In *AAAI Proc.*, pages 74–81, Austin, TX, August 2000.
19. David C Parkes and Lyle Ungar. Preventing strategic manipulation in iterative auctions: Proxy-agents and price-adjustment. In *AAAI Proc.*, pages 82–89, Austin, TX, August 2000.
20. Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
21. Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 256–262, Washington, D.C., July 1993.
22. Tuomas Sandholm. eMediator: A next generation electronic commerce server. In *Proceedings of the Fourth International Conference on Autonomous Agents (AGENTS)*, pages 73–96, Barcelona, Spain, June 2000. Early version appeared in the AAAI-99 Workshop on AI in Electronic Commerce, Orlando, FL, pp. 46–55, July 1999, and as a Washington University, St. Louis, Dept. of Computer Science technical report WU-CS-99-02, Jan. 1999.
23. Tuomas Sandholm. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129, 2000. Special Issue on Applying Intelligent Agents for Electronic Commerce. A short, early version appeared at the Second International Conference on Multi–Agent Systems (ICMAS), pages 299–306, 1996.
24. Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, January 2002. First appeared as an invited talk at the First International Conference on Information and Computation Economies, Charleston, SC, Oct. 25–28, 1998. Extended version appeared as Washington Univ., Dept. of Computer Science, tech report WUCS-99-01, January 28th, 1999. Conference version appeared at the International Joint Conference on Artificial Intelligence (IJCAI), pp. 542–547, Stockholm, Sweden, 1999.

25. Tuomas Sandholm and Subhash Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 90–97, Austin, TX, 2000. Extended version with proofs at www.cs.cmu.edu/ ˜ sandholm/bidbranching.aaai00.extended.ps.
26. Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1102–1108, Seattle, WA, 2001.
27. W Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.
28. Peter R Wurman and Michael P Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 21–29, Minneapolis, MN, October 2000.