Iterative Dutch Combinatorial Auctions

Shantanu Biswas * (biswas@csa.iisc.ernet.in) and Y. Narahari (hari@csa.iisc.ernet.in)
Electronic Enterprises Laboratory,
Computer Science and Automation,
Indian Institute of Science.

Abstract.
The combinatorial auction problem can be modeled as a weighted set packing problem. Similarly the reverse combinatorial auction can be modeled as a weighted set covering problem. We use the set packing and set covering formulations to suggest novel iterative Dutch auction algorithms for combinatorial auction problems. We use generalized Vickrey auctions (GVA) with reserve prices in each iteration. We derive worst case bounds for these algorithms.

Keywords: Combinatorial auctions, iterative auctions, generalized Vickrey auctions, weighted set packing problem, weighted set covering problem, incentive compatibility, efficiency, e-selling, e-procurement.

Table of Contents

* This research is supported in part by IBM Research Fellowship awarded to the author by IBM India Research Lab, New Delhi.

1. Introduction

Combinatorial auctions are auctions where bidders are allowed to submit bid on combinations of items. These are also called combinational or bundle auctions. Combinatorial auctions can be used for economically efficient allocations of goods, services, tasks, resources, etc. when the agents' valuations for bundles of items are not additive. The sub-additivity and super-additivity arises because some items may be substitutes, and others may be complementary. Combinatorial auctions have been suggested for many auction scenarios such as spectrum licenses, pollution permits, airport landing slots, computational resources, online procurement, and others (de Vries and Vohra, 2003; Narahari and Dayama, 2003).

Combinatorial auctions have major economic advantages but they are computationally complex. There has been a recent surge of interest in developing combinatorial auction algorithms (Rothkopf et al., 1998; Wurman and Wellman, 2000; Parkes, 2001; Lehmann et al., 2001; Kalagnanam and Parkes, 2003). Implementation of combinatorial auctions faces many challenges like succinctly representating various bids, efficient algorithms for solving the resulting NP-hard problems, and game theoretic issues of bidders' strategies and equilibria (Lehmann et al., 2001; Nisan, 2000). It has been shown that unless the underlying allocation problems are solved optimally the combinatorial auctions do not yield economically efficient solutions. This has led to a new field of research termed by some authors as algorithmic mechanism design. This field deals with interplay of algorithmic and game theoretic considerations (Papadimitriou, 2001; Nisan, 1999). The combinatorial auctions can be modeled as a weighted set packing problem (Bikhchandani and Ostroy, 2001).

In many market scenarios, e.g. in procurement, there is a buyer who wants to obtain some goods at the lowest possible cost, and a set of sellers who wish to sell different goods. A buyer can conduct a reverse auction (Sandholm et al., 2001; Gallien and Wein, 2000) to procure the bundle of goods. The pricing of bundles in reverse auctions can be sub-additive because the sellers may give volume discounts for bundles. As in the case of forward auction there are two common approaches to reverse auction:

1. Buyers post how much they are willing to pay for an item or a bundle and sellers respond with a bid.

2. Buyers identify the items or bundles that they are interested in purchasing without any indication of price. The interested sellers then submit their bids for various bundles.

We show that the reverse auction for procurement can be formulated as a weighted set covering problem.

Generalized Vickrey auction (GVA) or the Vickrey-Clarke-Groves (VCG) mechanism (Clarke, 1971; Groves, 1973) applied to combinatorial auctions generalizes the second price auction proposed by Vickrey (Vickrey, 1961) for single item auctions. This is a truthful or an incentive compatible mechanism for combinatorial auctions. However there are two problems with this scheme.

1. GVA is not budget balanced i.e. may yield low revenue for the seller.

2. GVA requires optimal solution of the allocation problem which is NP-hard.

Classical Dutch auctions (Wolfstetter, 1999) which are decreasing price auctions have been proposed for both single item and multi-unit homogeneous items. We suggest iterative Dutch auction schemes to reduce the complexity of these two problems of GVA. In our schemes:

– We know GVA is not budget balanced. And setting the reserve prices for the items is difficult because the agents bid for bundles instead of individual items. In our iterative dutch mechanisms the reserve prices for items are a natural outcome in each iteration.

– The second problem (i.e. GVA is NP-hard) depends on the size of the input. Therefore dividing the one shot GVA into smaller GVAs in each iteration significantly reduces the time to solve the problem. But the overall solution obtained may not be optimal. We show that the solutions obtained using these iterative schemes lie within provable worst case bounds. We conduct numerical experiments to show that in general the solutions are much better than the theoretical bounds and the allocations are very close to a Pareto efficient allocation is realized in these algorithms.

1.1. Contributions and Paper Outline

Our contributions in this paper are:

– We use the set covering and packing formulations to devise novel iterative Dutch auction schemes for combinatorial auctions.

– We also prove worst case bounds for these iterative auction algorithms.

In Section 2, we review the relevant literature. We present the integer programming formulation of forward and reverse combinatorial auctions. We discuss the greedy algorithms for these problems which give the best known bounds. We also discuss various aspects of generalized Vickrey auctions (GVA).

In Section 3, we present our iterative Dutch auction algorithms. We prove the bounds for these algorithms in Section 4. We also discuss the results of the numerical experiments that we carried out in Section 4. In Section 5 we conclude and discuss the scope of future work.

2. Review of Relevant Work

2.1. Integer Programming Formulation of Combinatorial Auction Problem

The general case of a combinatorial auction problem (Bikhchandani and Ostroy, 2001; de Vries and Vohra, 2003) can be stated as follows.

- An auctioneer (or a seller) wants to sell a set M of distinct objects.
- There are N bidders (or buyers) who are interested in buying the entire set or some subsets of M.
- The auctioneer or the seller wants to maximize the revenue earned.
- We assume XOR bidding language (Nisan, 1999) is used i.e. each bidder receives only one subset.

The seller wants to sell as much as possible while trying to maximize the revenue. This is a weighted set packing problem and can be formulated as an optimization problem.

- For every subset $S \in M$, let $v_j(S)$ be the value agent $j \in N$ assigns to consuming S.
- Let

$$y(S,j) = \begin{cases} 1 & \text{if the bundle } S \in M \text{ is allocated} \\ & \text{to agent } j \in N \\ 0 & \text{otherwise} \end{cases}$$

The optimization problem, denoted by (**IP**) is

$$\begin{aligned} V^* &= \max \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \\ s.t. \ & \sum_{S \ni i} \sum_{j \in N} y(S,j) \leq 1 \ \ \forall i \in M \\ & \sum_{S \subseteq M} y(S,j) \leq 1 \ \ \forall j \in N \\ & y(S,j) = 0,1 \ \ \forall S \subseteq M, \forall j \in N \end{aligned}$$

The first constraint ensures that overlapping sets of goods are never assigned. The second constraint ensures that no bidder receives more than one subset.

2.2. Integer Programming Formulation for Reverse Combinatorial Auction

In this problem we have a single buyer and multiple sellers. The buyer tries to procure from the sellers who quote the least prices. The buyer has to procure at least the required set while minimizing the procurement cost. Therefore the procurement problem becomes a weighted set covering problem. This is different from the forward combinatorial auction (i.e. packing problem) where we were trying to sell as much as possible.

The general case of a reverse combinatorial auction problem can be stated as follows.

- A buyer wants to buy a set M of distinct objects.
- There are N bidders who are interested in selling the entire set or some subsets of M.
- The auctioneer or the buyer wants to minimize the procurement cost.
- We assume XOR bidding language (Nisan, 1999) is used i.e. the buyer buys at most one subset from any bidder.

The problem can be formulated as

$$\begin{aligned} V^* &= \min \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \\ s.t. & \sum_{S \ni i} \sum_{j \in N} y(S,j) \geq 1 \ \ \forall i \in M \\ & \sum_{S \subseteq M} y(S,j) \leq 1 \ \ \forall j \in N \\ & y(S,j) = 0,1 \ \ \forall S \subseteq M, \forall j \in N \end{aligned}$$

where

- $v_j(S)$ is the valuation of the set S to seller j i.e. the seller j is willing to sell the set S for $v_j(S)$.
- $$y(S,j) = \begin{cases} 1 & \text{if the bundle } S \in M \text{ is allocated} \\ & \text{to agent } j \in N \\ 0 & \text{otherwise} \end{cases}$$

The first constraint ensures that at least the required set of goods is procured. The second constraint ensures that the buyer buys no more than one subset from any seller. And the objective is to minimize the total cost.

2.3. Minimum Weight Set Covering Problem

We discussed in the Subsection 2.2 that the reverse combinatorial auction can be modeled as a minimum weight set covering problem. The minimum weight set cover problem in its general form can be stated as

Instance:	A set system $(U, \mathcal{S})$ with $\bigcup_{S \in \mathcal{S}} = U$, weights $c : \mathcal{S} \to \mathbb{R}_+$.
Task:	Find a minimum weight set cover of $(U, \mathcal{S})$, i.e. a sub-family $\mathcal{R} \subseteq \mathcal{S}$ such that $\bigcup_{R \in \mathcal{R}} = U$.

Chvátal (Chvátal, 1979) gave a greedy algorithm for this problem. This algorithm motivates our iterative reverse Dutch auction described in Section 3.1. The greedy algorithm of Chvátal can be stated as follows.

A Greedy Algorithm *(*Chvátal, 1979*)*:

1. Set $\mathcal{R} := \phi$ and $W := \phi$.
2. While $W \neq U$ do:
   Choose a set $R \in \mathcal{S} \backslash \mathcal{R}$ for which $\frac{c(R)}{|R \backslash W|}$ is minimum.
   Set $\mathcal{R} := \mathcal{R} \cup R$ and $W := W \cup R$.

Theorem 1 *(*Chvátal, 1979*)* For any instance $(U, \mathcal{S}, c)$ of the minimum weight set cover problem, the greedy algorithm for set cover finds a set cover whose weight is at most $H(r)OPT(U, \mathcal{S}, c)$, *where* $r := max_{S \in \mathcal{S}} |S|$ *and* $H(r) = 1 + \frac{1}{2} + \cdots + \frac{1}{r}$.

This is the best known bound for the weighted set packing problem. There are some negative results which suggest that this may be the best possible bound. Raz and Safra (Raz and Safra, 1997) have shown that there exists a constant $c > 0$ such that, unless $P = NP$, no approximation ratio of $c\ log|U|$ (where $|U|$ = size of the set U) can be achieved. Also Feige (Feige, 1998) proved that approximation ratio of $c\ log|U|$ cannot be achieved for any $c < 1$ unless each problem in NP can be solved in $O(n^{O(loglog(n))})$ time.

2.4. Maximum Weight Set Packing Problem

In Subsection 2.1 we discussed that the forward combinatorial auction can be modeled as a maximum weight set packing problem. The maximum weight set packing problem in its general form can be stated as

Instance: A set system $(U, \mathcal{S})$ with $\bigcup_{S \in \mathcal{S}} = U$, weights $c : \mathcal{S} \to \mathbb{R}_+$.
Task: Find a maximum weight set packing of $(U, \mathcal{S})$, i.e. find a maximum weight sub-family $\mathcal{R} \subseteq \mathcal{S}$ whose elements are pairwise disjoint.

A greedy algorithm for this problem was given by (Lehmann et al., 2001). Our iterative forward Dutch auction described in Section 3.2 is motivated by this algorithm.
A Greedy Algorithm *(*Lehmann et al., 2001*)*:
The algorithm is executed in two phases.

1. In the first phase, the bids are sorted by some criterion i.e. a norm is defined and the bids are sorted in decreasing order following this norm.

2. In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say $< S, V(S) >$ is granted, i.e. the set S is allocated at price $V(S)$ and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies, i.e. does not grant, the bid.

Theorem 2: *(*Lehmann et al., 2001*):* The greedy allocation scheme with norm $\frac{V(S)}{|S|}$ approximates the optimal allocation within a factor of $|M|$, where M is the set of goods to be allocated.

There are better known bounds. In fact (Lehmann et al., 2001) have shown that the best known bound for the packing problem is a factor of $\sqrt{|M|}$ which can be achieved by a greedy allocation scheme with norm $\frac{V(S)}{\sqrt{|S|}}$.

2.5. Generalized Vickrey Auction

Generalized Vickrey auction (GVA) or the Vickrey-Clarke-Groves (VCG) mechanism (Clarke, 1971; Groves, 1973) applied to combinatorial auctions generalizes the second price auction proposed by Vickrey (Vickrey, 1961). GVA maximizes the sum of the declared utilities which are the true valuations of the bidders (incentive compatibility). Therefore the allocation maximizes the social welfare. In a quasi-linear setting this is equivalent to Pareto optimality. In other words, GVA assigns goods efficiently i.e. puts the goods in the hands of the bidder who values it most.

GVA can be described for a auction as follows (MacKie-Mason and Varian, 1995; Varian, 2000):

1. Each agent a reports a utility function $r_a(\cdot)$.

2. The planner computes

   $$x^* = \arg\max \sum_a r_a(x)$$

   subject to

   $$Feasibility\ Constraints$$

   and assigns action x_a^* to agent $a = 1, ..., A$. Then compute

   $$W_{-a}(x^*) = \sum_{b \neq a} r_b(x^*)$$

   which is the total valuation of all agents other than a according to their reported utility functions.

3. Agent a receives payoff

   $$u_a(x^*) - [G_a(r_{-a}) - W_{-a}(x_a^*)]$$

   where

   $$G_a(r_{-a}) = \max_x \sum_{b \neq a} r_b(x_{-a})$$

   subject to

   $$Feasibility\ Constraints$$

GVA is a truthful or an incentive compatible mechanism for combinatorial auctions. It is a dominant strategy for each agent to report his or her true utility function. To see this, note first that a necessary condition for the maximization in item (2) is that x^* maximizes $r_a(x) + W_{-a}(x_a)$. Agent a's true payoff is given in item (3). It follows that agent a will maximize his or her payoff by setting $r_a(x) = u_a(x)$. The set of actions taken will then be the actions that maximize the sum of the true utility functions.

This is the "second-price" analogue to the original Vickrey auction. Each agent is charged the total social surplus that would be possible if that agent did not participate in the auction at all. The result, then, is that the net payoff received by agent a is the net increment in total surplus that his participation creates. i i

However there are two problems with this scheme.

1. GVA may not be budget balanced i.e. may yield low revenue for the seller in forward auction or very high price for the buyer in the reverse auction.

2. GVA requires optimal solution of N+1 allocation problems (where N is the number of agents) which are NP-hard.

We know GVA is not budget balanced. The first problem (i.e. GVA is not budget balanced) can be mitigated by introducing reserve prices. Ausubel and Cramton (Ausubel and Cramton, 1999) prove that truthful bidding is a dominant strategy in case of private value auctions and an ex post equilibrium in case of interdependent values auctions. They also show that truth telling remains an ex post equilibrium in auction-plus-resale game, as long as resale game satisfies individual rationality. But setting the reserve prices for the items in combinatorial auctions is difficult because the agents bid for bundles instead of individual items.

The second problem has led to many approximate solution schemes and interesting auction algorithms (Wurman and Wellman, 2000; Parkes, 2001). Since the allocation problem is NP-hard it can be solved only approximately. And GVA may not be Pareto efficient if the allocation problem is not solved optimally (Lehmann et al., 2001).

2.6. Dutch Auctions

A Dutch auction is characterized by its decreasing price mechanism. The auction starts at a relatively high price and repeatedly decreases the price until a price announced by the auctioneer is accepted by one of the auction participants. The auction is then terminated and the bidder wins the auction. The Dutch auction got its name from the Dutch flower auction, where flowers are sold to traders. Dutch auction has been used traditionally for selling single objects such as works of art or single lots of a good such as cut flowers, fish, etc. This type of auction is also characterized by its speed. Usually the auctions are very short so that a lot of merchandise can be sold.

Dutch Auction is strategically equivalent to first price auction. Hansen (Hansen, 1988) showed that first-price auctions lead to a higher expected price in price dependent demands for multi-unit auctions, so that revenue equivalence breaks down in this case. Multi unit generalizations of Dutch and English auctions have been studied by McCabe et al (McCabe et al., 1980). But there has been no studies of dutch auctions for combinatorial auctions.

2.7. Our Work

We suggest iterative Dutch auction schemes in Section 3 to reduce the complexity of these two problems of GVA. In our schemes:

– We know GVA is not budget balanced. And setting the reserve prices for the items is difficult because the agents bid for bundles instead of individual items. In our iterative dutch mechanisms the reserve prices for items are a natural outcome in each iteration.

– The second problem (i.e. GVA is NP-hard) depends on the size of the input. Therefore dividing the one shot GVA into smaller GVAs in each iteration significantly reduces the time to solve the problem. But the overall solution obtained may not be optimal. We show that the solutions obtained using these iterative schemes lie within provable worst case bounds.

3. Iterative Dutch Auction Schemes

3.1. Iterative Reverse Dutch Auction (IRDA)

Classical (forward) Dutch auctions are decreasing auctions which have been conducted for both single item and multi-unit homogeneous items. In the single unit Dutch auction the auctioneer begins at a high price and incrementally lowers it until some bidder signals acceptance. Similarly in the multi-unit case the price is incrementally reduced till all the items are sold or the seller's reserve price is reached.

In the reverse auction for procurement the buyer tries to procure a bundle of items. The buyer starts with a low initial willingness to pay (say equal to zero) and keeps on incrementally increasing the willingness to pay until the total bundle is procured or the budget limit is reached. The buyer has a procurement budget. But this total budget cannot be divided linearly into budget for each item because of the complementarities involved. The buyer may value the entire bundle at a certain price but the value of a partial bundle may be much less.

Our iterative mechanism consists of multiple bidding rounds denoted by $t \in \mathbb{Z}_+$ ($t = 0$ is the initial round). The buyer sets $W(B_t)$, maximum willingness to pay for the remaining bundle B_t to be procured in round t. The pricing of items is not linear, therefore the cost of the allocated bundles cannot be divided into price of individual items. Therefore we calculate p_t, the average willingness of the buyer to pay for each item in round t

$$p_t = \frac{W(B_t)}{|B_t|}, \text{ where } B_t \neq \phi$$

.
Let the payment made by the buyer for the subset S_t in iteration t be $V^*(S_t)$. The average price paid by the buyer for each item procured is

$$v_t = \frac{V^*(S_t)}{|S_t|}$$

. We neglect the iterations in which no items are procured. The average price paid by the buyer for each item in iteration $t-1$ is v_{t-1}. We set the the reserve price $R(S_t)$ of the seller for any bundle S_t in iteration t to $|S_t|v_{t-1}$.

We use GVA with reserve prices in each iteration. Therefore we have the following two cases for the payment $V^*(S_t)$ made by the buyer for the subset S_t in iteration t:

1. If $R(S_t) \leq Vickrey\ Price\ for\ the\ set\ S_t < W(B_t)$, then

$$V^*(S_t) = Vickrey\ Price\ for\ the\ set\ S_t$$

2. If $Vickrey\ Price\ for\ the\ set\ S_t \geq W(B_t)$, then we have the budget imbalance case. Therefore we set

$$V^*(S_t) = R(S_t)$$

Since we use GVA with reserve prices in each iteration, we shall have to solve NP-hard allocation problems in each iteration. But the problem size in each iteration is much smaller than the complete problem. The time taken to solve the smaller GVAs in each iteration is much less than the time taken to solve the complete problem, since the solution time grows exponentially with size of the problem.

Notation

t	=	Iteration number
B_t	=	Bundle remaining to be procured in iteration t
S_t	=	Set procured in iteration t
$V^*(S_t)$	=	Actual buying price of the set S_t in iteration t
v_t	=	Average buying price of each item bought in iteration t
p_t	=	Average Price of each item set by the auctioneer in iteration t
$W(B_t)$	=	Maximum procurement price set by the auctioneer for the bundle B_t in iteration t
$R(S_t)$	=	Reserve price of sellers for any bundle S_t in iteration t
ε	=	Price decrement per item in every iteration

Therefore the integer programming formulation of the GVA problem with reserve prices in iteration t becomes

$$\begin{aligned}
V^*(S_t) &= \min \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S, j) \\
s.t. \quad & \sum_{S \ni i} \sum_{j \in N} y(S, j) \geq 1 \ \ \forall i \in M \\
& \sum_{S \subseteq M} y(S, j) \leq 1 \ \ \forall j \in N \\
& v_j(S) y(S, j) \geq |S| v_{t-1} \ \ \forall S \subseteq M, \forall j \in N
\end{aligned}$$

$$\sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S, j) \leq W(B_t) \ \ \forall S \subseteq M$$
$$y(S, j) = 0, 1 \ \ \forall S \subseteq M, \forall j \in N \quad (1)$$

3.1.1. *IRDA Algorithm*
The IRDA algorithm can be described as follows:

1. Suppose the buyer's initial willingness to pay for the entire bundle B_0 is zero i.e. $W_(B_0) = 0$. Therefore the willingness to pay for each item is also zero i.e. $p_0 = 0$. Since no sellers are likely to be interested to bid at this price, therefore
$$V^*(S_0) \ = v_0 \ = \ 0$$

2. Increment the average willingness to pay for each item by ε to $p_1 = v_0 + \varepsilon$. This actually means that the buyer's willingness to pay for the bundle B_1 is changed to $W(B_1) = |B_1| \times p_1$. We assume that the increment ε in every iteration is constant. The reserve price of any bundle S_t for the sellers becomes $|S_t|v_0$.

3. Solve the allocation problem if there are any bids i.e. for iteration $t \ = \ 1$ solve the Eq. 1 and calculate v_1. This is again a combinatorial optimization problem. But this is much smaller than the complete problem.

4. Allocate the subsets to the winners. Remove the allocated items from the set to be procured and increment the average willingness to pay for each item to $p_2 = v_1 + \varepsilon$, i.e. the maximum willingness of the buyer to pay for the remaining bundle B_2 is $W(B_2) \ = \ |B_2| \times p_2$. The new reserve price of any bundle S_t of items for the sellers is $|S_t|v_1$.

5. Go to step 3 and repeat until the buyer can procure the entire bundle or the upper limit i.e. the total procurement budget is reached . In any iteration t the following condition should be satisfied:

$$\text{total procurement budget} \ \geq \ W(B_t) + \sum_{i=0}^{t-1} V^*(S_i)$$

3.2. Iterative Forward Dutch Auction (IFDA)

In our combinatorial version of iterative forward Dutch auction the seller starts with a high initial price and keeps on decreasing the price until the total bundle is sold. As in the reverse mechanism, in the iterative forward Dutch auction the bidding rounds are denoted by $t \in \mathbb{Z}_+$ ($t = 0$ is the initial round).

The sellers provides $W(B_t)$, total ask for the remaining bundle B_t to be sold in round t. We cannot divide the ask price of the bundle into ask prices of individual items because of non linear pricing. The average ask price of each item in round t is p_t. Therefore we calculate p_t, the average ask price of the seller for each item in round t.

$$p_t = \frac{W(B_t)}{|B_t|}, \text{ where } B_t \neq \phi$$

Let the payment earned by the seller for the subset S_t in iteration t be $V^*(S_t)$. The average selling price for each item sold is

$$v_t \; = \; \frac{V^*(S_t)}{|S_t|}$$

. We neglect the iterations in which no items are procured. The average selling price each item in iteration $t-1$ is therefore v_{t-1}. The reserve price of the seller in iteration t for the remaining bundle B_t is $W(B_t)$. Also set the the maximum willingness of buyers $M(S_t)$ to pay for any bundle S_t in iteration t to $|S_t|v_{t-1}$.

We use GVA with reserve prices in each iteration. Therefore we have the following two cases for the actual selling price $V^*(S_t)$ for the subset S_t in iteration t:

1. If $Vickrey\ Price\ for\ the\ set\ S_t > 0$, then

$$V^*(S_t) = Vickrey\ Price\ for\ the\ set\ S_t$$

2. If $Vickrey\ Price\ for\ the\ set\ S_t = 0$, then we have the budget imbalance case. Therefore we set

$$V^*(S_t) = M(S_t)$$

Notation

t	=	Iteration number
B_t	=	Bundle remaining to be sold in iteration t
(S_t)	=	Set sold in iteration t
$V^*(S_t)$	=	Actual selling price of the set S_t in iteration t
v_t	=	Average selling price of each item bought in iteration t
p_t	=	Average price of each item set by the auctioneer in iteration t
$W(B_t)$	=	Minimum ask price set by the auctioneer for the bundle B_t in iteration t
$M(S_t)$	=	Maximum willingness of buyers to pay for any bundle S_t in iteration t
ε	=	Price increment per item in every iteration

Therefore the integer formulation of the GVA problem with reserve prices in iteration t becomes

$$\begin{aligned} V^*(S_t) \ &= \max \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \\ s.t. \quad & \sum_{S \ni i} \sum_{j \in N} y(S,j) \leq 1 \ \ \forall i \in M \\ & \sum_{S \subseteq M} y(S,j) \leq 1 \ \ \forall j \in N \\ & \sum_{j \in N} \sum_{S \subseteq M} v_j(S) y(S,j) \geq W(B_t) \ \ \forall S \subseteq M \\ & v_j(S) y(S,j) \leq |S| v_{t-1} \ \ \forall S \subseteq M, \forall j \in N \\ & y(S,j) = 0,1 \ \ \forall S \subseteq M, \forall j \in N \end{aligned} \tag{2}$$

3.2.1. *IFDA Algorithm*

We can now describe the IFDA algorithm as follows:

1. Let the seller's initial ask price for each item be $p_0 = P$, where P is a very large integer i.e. the seller's initial ask price for the bundle B_0 is $|S_0| \times P$. At this very high price no bids are likely to be submitted. Therefore $V^*(S_0) = 0$ and $v_0 = p_0$. If there are any bids then we solve for $V^*(S_0)$

2. Decrease the ask price for each item by ε to $p_1 = v_0 - \varepsilon$. Therefore the seller's ask price for the bundle B_1 is $W(B_1) = p_1 \times |B_1|$. The maximum willingness of buyers to pay for any bundle S_t become $|S_t| \times v_0$.

3. If there are some bids then solve the allocation problem for iteration $t = 1$ given in Eq. 2. This is again a combinatorial optimization problem. But this is much smaller than the original overall problem.

4. Allocate the subsets to the winners. Update the set to be procured and the decrease the seller's ask prices to $p_2 = v_1 - \varepsilon$ i.e. $W(B_2) = p_2 \times |B_2|$. The maximum willingness of buyers to pay for any bundle S_t become $|S_t| \times v_1$.

5. Go to step 3 and repeat until we can sell the entire bundle or we reach the lower limit i.e. seller's actual reserve price for the entire bundle. In any iteration t the following condition should be satisfied

$$\text{Reserve Price for the entire bundle} \ \leq \ |S_t| p_t + \sum_{i=0}^{t-1} V^*(S_i)$$

4. Bounds for the Iterative Dutch Auction Algorithms

4.1. Convergence of the Algorithms

Lemma *(Termination of the Algorithm):* Both algorithms (IRDA and IFDA) terminate in finite number of steps.
Proof: The proof is quite straightforward. We know every monotonic increasing (or decreasing) sequence which is bounded above (or below) converges. We have two cases here:

1. IRDA: We have the sequence

$$p_0, p_1, p_2, \dots$$

$$p_0 = v_0, p_1 = v_0 + \varepsilon, p_2 = v_1 + \varepsilon, \dots$$

   where:

$$\varepsilon > 0,$$

$$v_0 \leq v_1 \leq v_2 \leq \cdots, \textit{ and}$$

$$v_t \leq \frac{Total\ Procurement\ Budget\ - \sum_{i=0}^{t-1} V^*(S_1)}{|B_t|}$$

   Therefore this sequence is monotonic increasing and bounded above. Hence it converges.

2. IFDA: We have the sequence

$$p_0, p_1, p_2, \dots$$

$$p_0 = v_0, p_1 = v_0 - \varepsilon, p_2 = v_1 - \varepsilon, \dots$$

   where:

$$\varepsilon > 0,$$

$$v_0 \geq v_1 \geq v_2 \geq \cdots, \textit{ and}$$

$$p_t \geq \frac{Reserve\ price\ for\ the\ entire\ bundle\ - \sum_{i=0}^{t-1} V^*(S_1)}{|B_t|}$$

   Therefore this sequence is monotonic decreasing and bounded below. Hence it converges.

Thus both the algorithms terminate in finite number of stages.

Theorem 3 *(Upper Bound for the Reverse Dutch Auction):* The upper bound for the IRDA is

$$(1+\frac{1}{2}+...+\frac{1}{r})V^*, \text{ where}$$
$$V^* = \text{ Optimal Solution of the overall problem, and}$$
$$r = max_{S\in M}|S|$$

Proof:

Let the subsets allocated in each iterations be $S_1, S_2, ..., S_K$, where there are K iterations. We can ignore the iterations in which there was no allocation without any loss of generality.

We have defined the reserve prices in iteration t as

$$|S_t| \times v_{t-1} \leq V^*(S_t) = v_t \times |S_t|$$

$$\Rightarrow v_{t-1} \leq \frac{V^*(S_t)}{|S_t|} = v_t$$

$$\Rightarrow \frac{V^*(S_{t-1})}{|S_{t-1}|} = v_{t-1} \leq \frac{V^*(S_t)}{|S_t|} = v_t$$

$$\Rightarrow \frac{V^*(S_{t-1})}{|S_{t-1}|} \leq \frac{V^*(S_t)}{|S_t|}$$

$$\Rightarrow \frac{V^*(S_1)}{|S_1|} \leq \frac{V^*(S_2)}{|S_2|} \leq ... \frac{V^*(S_k)}{|S_k|}$$

In other words this is the same as the greedy algorithm of Chvátal (Section 2.3) when all the bids are given at once. This is true under the assumption that bidding is truthful for both one shot GVA and IRDA. We know GVA is incentive compatible. Also we use GVA with reserve prices in each iteration of IRDA. Thus the upper bound for the reverse Dutch algorithm is

$$(1+\frac{1}{2}+...+\frac{1}{r})V^*, \text{ where } r = max_{S\in M}|S|$$

4.2. LOWER BOUND FOR THE FORWARD DUTCH AUCTION

Theorem 4 *(Lower Bound for the Forward Dutch Auction):* The lower bound for the IFDA is

$$\begin{aligned}&\frac{1}{|M|}V^*, \text{ where}\\ V^* &= \text{Optimal solution of the overall problem, and}\\ |M| &= \text{total number items to be sold.}\end{aligned}$$

Proof: The proof is similar to the proof of theorem 3. We get the following inequalities in this case
We know

$$W(B_t) = W(B_{t-1}) - \varepsilon$$

and

$$W(B_{t+1}) = W(B_t) - \varepsilon$$

Also

$$|S_{t+1}|v_t \geq V^*(S_{t+1}) = |S_{t+1}|v_{t+1}$$

$$\Rightarrow v_{t+1} \leq v_t$$

$$\Rightarrow \frac{V^*(S_{t+1})}{|S_{t+1}|} \leq \frac{V^*(S_t)}{|S_t|}$$

$$\Rightarrow \frac{V^*(S_1)}{|S_1|} \geq \frac{V^*(S_2)}{|S_2|} \geq \ldots \frac{V^*(S_k)}{|S_k|}$$

This is same as the greedy algorithm of (Lehmann et al., 2001) (Section 2.4) with norm $\frac{V(S)}{|S|}$.
Therefore we get the lower bound as

$$\frac{1}{|M|}V^*$$

4.3. NUMERICAL EXPERIMENTS

We have shown the worst case (lower or upper) bounds for these iterative Dutch auction algorithms. We have run our algorithms on some of the test cases suggested by various authors (Sandholm, 1999; Boutilier et al., 1999; de Vries and Vohra, 2003; Kevin Leyton-Brown, 2000). We have used CPLEX^{TM} 8.0 to solve the various instances of GVA with reserve prices. The preliminary results suggest that the average case performance of these approximate schemes are much better than the proven lower or upper bounds. Our simulation results indicate that the proposed algorithms perform as well as the GVA in terms of the social surplus and the seller's revenue or buyer's procurement cost. Our algorithms cannot guarantee to obtain a Pareto efficient allocation. But the simulation results suggest that allocations obtained by these algorithms are very close to a Pareto efficient allocations. Also in our algorithms, the required computational costs for an auctioneer is much lower than the cost of performing the GVA. More experiments are needed to study the performance of our iterative schemes on various distribution of bids.

5. Conclusions and Future Work

The forward combinatorial auction problem is a set packing problem as shown by various authors (Bikhchandani and Ostroy, 2001). We have shown that the problem of reverse combinatorial auction for procurement can be modeled as a weighted set covering problem. We use these set packing and covering formulations to devise iterative Dutch auction mechanisms. These iterative mechanisms use GVA with reserve prices in each iteration. We have shown the worst case (lower or upper) bounds for these iterative Dutch auction algorithms. The average case performance of these approximate schemes seems to be better than these bounds. More experiments are needed to study the performance of our iterative schemes.

We have shown that our algorithms converge. But the rate of convergence of the algorithms depend on ε i.e. the bid increment in each iteration. If ε is large, the algorithm will converge very fast but the iterative problem will become almost the same as one shot GVA. On the other hand, if ε is very small we will have lots of iterations and the quality of solution may not be very good. Therefore we need to calculate the rate of convergence in terms of ε. We also need to extend our algorithms to the case of variable bid increments. We can use learning algorithms in case of the variable increments.

References

Ausubel, L. and P. Cramton: 1999, 'Vickrey Auctions with Reserve Pricing'. Technical report, Working Paper, University of Maryland.

Bikhchandani, S. and J. Ostroy: 2001, 'The Package Assignment Model'. Technical report, Working Paper, University of California, Los Angeles.

Boutilier, C., M. Goldszmidt, and S. B.: 1999, 'Sequential auctions for the allocation of resources with omplementarities'. In: *International Joint Conferences on Artificial Intelligence*.

Chvátal, V.: 1979, 'A greedy heuristic for the set cover problem'. *Mathematics of Operations Research* **4**, 233–235.

Clarke, E.: 1971, 'Multipart Pricing of Public Goods'. *Public Choice* **8**, 19–33.

de Vries, S. and R. V. Vohra: 2003, 'Combinatorial Auctions: A Survey'. *INFORMS Journal of Computing* **15**(1).

Feige, U.: 1998, 'A threshold of ln *n* for the approximating set cover'. *Journal of ACM* **45**, 634–652.

Gallien, J. and L. Wein: 2000, 'Design and Analysis of a Smart Market for Industrial Procurement'. Technical report, Working Paper, MIT, Cambridge, MA.

Groves, T.: 1973, 'Incentives in Teams'. *Econometrica* **41**, 617–631.

Hansen, R.: 1988, 'Auctions with endogenous quantity'. *Rand Journal of Economics* **19**, 44–58.

Kalagnanam, J. and D. Parkes: 2003, 'Auctions, Bidding, and Exchange Design'. In: Simchi-Levi, D. Wu, and Shen (eds.): *Supply Chain Analysis in the eBusiness Area*. Kluwer Academic Publishers.

Kevin Leyton-Brown, Mark Pearson, Y. S.: 2000, 'Towards a Universal Test Suite for Combinatorial Auction Algorithm'. In: *ACM Conference on Electronic Commerce, 2000*.

Lehmann, D., L. O'Callaghan, and Y. Shoham: 2001, 'Truth Revelation in Approximately Efficient Combinatorial Auctions'. Technical report, Working Paper, Hebrew University, Jerusalem.

MacKie-Mason, J. K. and H. R. Varian: 1995, 'Generalized Vickrey Auctions'. Technical report, University of Michigan.

McCabe, K. A., S. J. Rassenti, and V. L. Smith: 1980, 'Auction Institutional Design: Theory and Behavior of Simultaneous Multiple-Unit Generalizations of Dutch and English Auctions'. *The American Economic Review* **80**(5), 1276–1283.

Narahari, Y. and P. Dayama: 2003, 'Combinatorial Auctions for Electronic Business'. Technical report, Electronic Enterprises Laboratory, Department of Computer Science and Automation, Indian Institute of Science.

Nisan, N. Ronen, A.: 1999, 'Algorithmic Mechanism Design'. In: *31st ACM Symposium on the Theory of Computing*.

Nisan, N.: 2000, 'Bidding and Allocation in Combinatorial Auctions'. In: *ACM Conference on Electronic Commerce, 2000*.

Papadimitriou, C.: 2001, 'Algorithms, Games, and the Internet'. In: *33rd ACM Symposium on the Theory of Computing*.

Parkes, D.: 2001, 'Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency'. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Raz, R. and S. Safra: 1997, 'A sub constant error probability low degree test, and a sub constant error probability PCP characterization of NP'. In: *29th ACM Symposium on the Theory of Computing*. pp. 475–484.

Rothkopf, M. H., A. Pekec, and R. M. Harstad: 1998, 'Computationally manageable combinatorial auctions'. *Management Science* **44**, 1131–1147.

Sandholm, T.: 1999, 'An algorithm for optimal winner determination in combinatorial auctions'. In: *International Joint Conferences on Artificial Intelligence*.

Sandholm, T., S. Suri, A. Gilpin, and D. Levine: 2001, 'Winner determination in combinatorial auction generalizations'. In: *AGENTS-2001 Workshop on Agent-Based Approaches to B2B, Montreal, Canada*.

Varian, H.: 2000, 'Economic Mechanism Design for Computerized Agents'. Technical report, University of California, Berkeley.

Vickrey, W.: 1961, 'Counterspeculation, Auctions, and Competitive Sealed Tenders'. *Journal of Finance* **16**, 8–37.

Wolfstetter, E.: 1999, *Topics in Microeconomics*. Cambridge University Press, Cambridge, UK.

Wurman and Wellman: 2000, 'AkBA: A Progressive, Anonymous-Price Combinatorial Auction'. In: *ACM Conference on Electronic Commerce, 2000*.