A Decomposition Based Approach for Design of Supply Aggregation and Demand Aggregation Exchanges

Shantanu Biswas[1], Y. Narahari[1], and Anish Das Sarma[2]

[1] Electronic Enterprises Laboratory,
Computer Science & Automation,
Indian Institute of Science, Bangalore, India,
{biswas,hari}@csa.iisc.ernet.in
[2] Computer Science & Engineering,
Indian Institute of Technology, Bombay, India,
anish@cse.iitb.ac.in

Abstract. Combinatorial exchanges are double sided marketplaces with multiple sellers and multiple buyers trading with the help of combinatorial bids. The allocation and other associated problems in such exchanges are known to be among the hardest to solve among all economic mechanisms. In this paper, we study combinatorial exchanges where (1) the demand can be aggregated, for example, a procurement exchange or (2) the supply can be aggregated, for example, an exchange selling excess inventory. We show that the allocation problem in such exchanges can be solved efficiently through decomposition when buyers and sellers are single minded. The proposed approach decomposes the problem into two stages: a forward or a reverse combinatorial auction (stage 1) and an assignment problem (stage 2). The assignment problem in Stage 2 can be solved optimally in polynomial time and thus these exchanges have computational complexity equivalent to that of one sided combinatorial auctions. Through extensive numerical experiments, we show that our approach produces high quality solutions and is computationally efficient.

1 Introduction

1.1 Combinatorial Exchanges

Combinatorial exchanges are two sided combinatorial mechanisms involving multiple buyers and multiple sellers. Combinatorial exchanges (CEs) have major economic advantages due to the power of combinatorial bidding and highly expressive bidding languages. However, they are, at the same time notoriously complex from a computational angle. There are two main computational problems associated with CEs:

– Allocation problem (also called winner determination problem or trade determination problem): Choosing an optimal subset of bids so as to optimize

a chosen performance metric (such as revenue maximization, cost minimization, surplus maximization, etc.).
- Pricing problem: Determining the actual payments to be made by the buyers and actual payments to be made to the sellers, so as to induce truthful bidding by the buyers and sellers.

The above problems have been shown to be NP-hard [1, 2]. Researchers have therefore been seeking computationally efficient ways of solving these problems exactly or approximately.

Numerous industrial applications have been reported in the literature for combinatorial exchanges. These include: collaborative planning [3], logistics and transportation exchanges [4–6], bandwidth exchanges [7], steel exchanges (for example, `www.esteel.com`), supply chain formation [8], e-procurement exchanges [9, 10], stock exchanges, and exchanges for sale of excess inventory.

1.2 Combinatorial Exchanges with Aggregation

Our motivation in this paper is to study combinatorial exchanges where:

- the demand can be aggregated, for example, a procurement exchange where the buyers' demands can be aggregated,
- the supply can be aggregated, for example, an exchange selling excess inventory where the sellers' bids can be aggregated.

1.3 Contributions and Outline

We show that when agents are single minded the aggregated combinatorial exchanges can be decomposed into two separate stages.

- Stage 1: A combinatorial auction
- Stage 2: An assignment problem

Since the problem in Stage 2 can be solved exactly in polynomial time, therefore these exchanges have computational complexity equivalent to one sided combinatorial auctions. We show that the iterative Dutch auction schemes proposed in [11] can be extended to solve the problem in stage 1 in a computationally efficient way to obtain near optimal allocations.

The rest of the paper is organized as follows. Section 2 presents a review of relevant work to put our contributions in perspective. In Section 3, we present a mathematical formulation of the allocation problem for both demand aggregation and supply aggregation combinatorial exchanges and show that these exchanges can be decomposed into a two stage problem. In Section 4, we present generalized combinatorial Dutch auction mechanisms for solving the combinatorial auction problem in stage 1. In Section 5, we present numerical experiments. We summarize the contributions of the paper in Section 6.

2 Review of Relevant Work

Combinatorial exchanges (CE) are generalizations of combinatorial auctions. Also, CEs are multi-item exchanges, so they are more general than single item, multi-unit exchanges. Several survey papers have appeared on combinatorial auctions. These include the papers by de Vries and Vohra [2], and Narahari and Pankaj [12]. Similarly, there are several papers on single item, multi-unit exchanges, which are summarized in [1].

The paper by Smith, Sandholm, and Simmons [13] presents a design for a market maker to construct and clear a combinatorial exchange for trading single units of multiple items. Pankaj and Narahari [14] have applied a decomposition idea to solve electronic exchanges trading multiple units of a single homogeneous item. The paper by Kothari, Sandholm, and Suri [15] considers a multi-unit, multi-item combinatorial exchange where acceptance of partial bids is allowed. Parkes, Kalagnanam, and Eso [16, 17] design a combinatorial exchange that is approximately efficient and approximately truthful.

Iterative or indirect mechanisms are those where multiple rounds of bidding and allocation are conducted and the problem is solved in an iterative and incremental way [1]. Such an approach has several advantages: incremental information revelation, reduction in the complexity of the allocation problem, and practical appeal in industrial applications. There are several iterative mechanisms proposed for combinatorial auctions [1, 16]. The use of iterative mechanisms for clearing combinatorial exchanges has been proposed by Biswas [18]. The mechanisms proposed include auction mechanisms, Dutch auction mechanisms, and tâtonnement mechanisms.

Biswas and Narahari [11] have proposed the use of combinatorial Dutch auctions as an iterative mechanism for combinatorial auctions. They use the weighted set packing structure of forward combinatorial auctions to suggest an iterative forward Dutch auction algorithm for forward combinatorial auction problems, using generalized Vickrey auctions (GVA) with reserve prices in each iteration. They prove the convergence of the proposed algorithm and derive worst case bounds for the algorithm. Similarly, they use the set covering structure of reverse combinatorial auctions to suggest an iterative reverse Dutch auction algorithm for reverse combinatorial auction problems. They also show that the proposed algorithms produces near optimal quality solutions and are computationally efficient. In this current paper, we will be using these combinatorial Dutch auction algorithms in the first stage of solving aggregated combinatorial exchanges.

3 Formulation of the Exchange Problems

We consider two special cases of combinatorial exchanges: Demand aggregation combinatorial exchanges and supply aggregation combinatorial exchanges. We assume that the goods can be freely disposed, that is, the supply is greater than or equal to the demand.

3.1 Demand Aggregation Combinatorial Exchanges

Buyers submit their requirements, that is, the set of items and the quantities they want to procure. We assume that

- the buyers are single minded i.e. they are interested only in a single subset.
- the buyers submit atomic (all-or-nothing) bids i.e. they are interested only in a single set The buyers also submit their maximum willingness to pay for the set.

The exchange pools the demand and accepts bids from the interested sellers for the aggregated demand. The allocation problem can be formulated as follows. We give the notation used in Table 1.

Table 1. Notation used for the demand and supply aggregation exchange models

$M = \{1, 2, ..., m\}$	Set of seller agents
$i \in M$	Seller agent
$N = \{1, 2, ..., n\}$	Set of buyer agents
$j \in N$	Buyer agent
$K = \{1, 2, ..., l\}$	Set of items
$k \in K$	Any item
$G = \{S_1, S_2, ...\}$	Collection of all multi-sets over K.
$S \subseteq G$	Any subset of G
$a(S, k)$	Quantity of item k in the subset S
$v(S, i)$	Value of the subset S to seller agent i
$v(S, j)$	Value of the subset S to buyer agent j
$y(S, i)$	Boolean variable indicating whether or not seller i gets the set S
$y(S, j)$	Boolean variable indicating whether or not buyer j gets the set S

The integer programming formulation of the exchange problem is:

$$V = \max \left\{ \sum_{j \in N} \sum_{S \subseteq G} v(S, j) y(S, j) - \sum_{i \in M} \sum_{S \subseteq G} v(S, i) y(S, i) \right\}$$

$$s.t.$$

$$\sum_{S \subseteq G} y(S, i) \leq 1 \ \ \forall i \in M$$

$$\sum_{S \ni k} \sum_{j \in M} a(S, k) y(S, j) \leq \sum_{S \ni k} \sum_{i \in M} a(S, k) y(S, i) \ \ \forall k \ \ \forall S \subseteq G$$

$$y(S, i) = 0, 1 \ \ \forall S \subseteq G, \forall i \in M$$

$$y(S, j) = 0, 1 \ \ \forall S \subseteq G, \forall j \in N$$

Since the demand is aggregated we can decompose the above problem into two stages. In Stage 1, knowing the aggregated demands, we solve a combinatorial procurement problem to procure the demanded items at minimum cost by choosing from the bids submitted by the sellers. In Stage 2, the procured items are sold to the buyers so as to generate maximum revenue by choosing from the bids submitted by the buyers. This is described more formally below.

Stage 1: We have assumed that the buyers are single minded. Therefore, we can aggregate the buyers' demand. Let q_k be the aggregate demand of item k, where

$$q_k = \sum_{S \ni k} \sum_{j \in N} a(S,k) y(S,j)$$

The allocation problem for Stage 1 becomes:

$$\begin{aligned}
V_1 &= \min \sum_{i \in M} \sum_{S \subseteq G} v(S,i) y(S,i) \\
s.t. & \\
& \sum_{S \subseteq G} y(S,i) \leq 1 \quad \forall i \in M \\
& \sum_{S \ni k} \sum_{i \in M} a(S,k) y(S,i) \geq q_k \quad \forall k \quad \forall S \subseteq G \\
& y(S,i) = 0,1 \quad \forall S \subseteq G, \forall i \in M
\end{aligned}$$

This is precisely the formulation for reverse combinatorial auction for procurement as in [11] and in [7].

Stage 2: The problem in the second stage is to allocate to the buyers the items procured in the first stage to maximize the revenue of the exchange. The integer programming formulation of this stage is:

$$\begin{aligned}
V_2 &= \max \sum_{j \in N} \sum_{S \subseteq G} v(S,j) y(S,j) \\
s.t. & \\
& \sum_{S \ni k} \sum_{j \in N} a(S,k) y(S,i) \leq q_k \quad \forall k \quad \forall S \subseteq G \\
& y(S,j) = 0,1 \quad \forall S \subseteq G, \forall j \in N
\end{aligned}$$

We can immediately note that the above is an assignment problem and therefore can be solved exactly in polynomial time.

We have thus decomposed the allocation problem of a demand aggregation combinatorial exchange into a reverse combinatorial procurement auction followed by an assignment problem.

3.2 Supply Aggregation Combinatorial Exchanges

Here the sellers declare to the exchange their individual total supplies and corresponding reserve prices. We assume that

- the sellers are single minded.
- the sellers submit atomic asks. But this can be generalized to other bidding languages such as XOR or OR*.

Then the exchange asks the interested buyers to submit their bids for the aggregated supply.

The notation for the supply aggregation exchange problem is given Table 1. The exchange can now be formulated as given below.

$$
\begin{aligned}
V &= \max \left\{ \sum_{j \in N} \sum_{S \subseteq G} v(S,j) y(S,j) - \sum_{i \in M} \sum_{S \subseteq G} v(S,i) y(S,i) \right\} \\
s.t. & \\
& \sum_{S \subseteq G} y(S,j) \leq 1 \ \ \forall j \in N \\
& \sum_{S \ni k} \sum_{j \in N} a(S,k) y(S,j) \leq \sum_{S \ni k} \sum_{i \in M} a(S,k) y(S,i) \ \ \forall k \ \ \forall S \subseteq G \\
& y(S,i) = 0,1 \ \ \forall S \subseteq G, \forall i \in M \\
& y(S,j) = 0,1 \ \ \forall S \subseteq G, \forall j \in N
\end{aligned}
$$

Since the aggregated supply is known, we can decompose the above problem into two stages. In Stage 1, knowing the aggregated supply, we solve a combinatorial selling problem to sell the items to maximize revenue by choosing from the bids submitted by the buyers. In Stage 2, we procure the sold items from the sellers so as to minimize procurement cost by choosing from the bids submitted by the sellers. This is described more formally below.

Stage 1: We have assumed that the sellers are single minded. Therefore, we can aggregate the total supply from all the suppliers. Let q_k be the aggregated supply of item k, where

$$
q_k = \sum_{S \ni k} \sum_{i \in M} a(S,k) y(S,i)
$$

The allocation problem for Stage 1 becomes:

$$
\begin{aligned}
V_1 &= \max \sum_{j \in N} \sum_{S \subseteq G} v(S,j) y(S,j) \\
s.t. & \\
& \sum_{S \subseteq G} y(S,j) \leq 1 \ \ \forall j \in N
\end{aligned}
$$

$$\sum_{S \ni k} \sum_{j \in N} a(S,k) y(S,j) \leq q_k \ \ \forall k \ \ \forall S \subseteq G$$
$$y(S,j) = 0,1 \ \ \forall S \subseteq G, \forall j \in N$$

This is precisely the formulation for forward combinatorial auction discussed in [11] and in [19, 2].

Stage 2: The problem in the second stage is to allocate to the sellers the items sold to the buyers in the first stage. The integer programming formulation of the stage stage is:

$$V_2 = \min \sum_{j \in N} \sum_{S \subseteq G} c(S,j) y(S,j)$$
$$s.t.$$
$$\sum_{S \ni k} \sum_{j \in N} a(S,k) y(S,j) \geq q_k \ \ \forall k \ \ \forall S \subseteq G$$
$$y(S,j) = 0,1 \ \ \forall S \subseteq G, \forall j \in N$$

Note that the above is an assignment problem which can be solved exactly in polynomial time.

We have thus decomposed the aggregation combinatorial exchanges into a combinatorial auction followed by an assignment problem. Different mechanisms have been proposed for combinatorial auction problems. We generalize the iterative Dutch auctions suggested in [11] for the aggregation exchange problems.

4 Combinatorial Dutch Auctions for Aggregation Exchanges

We can extend the iterative Dutch auctions suggested in [11] for the aggregation exchanges as follows.

4.1 Reverse Combinatorial Dutch Auction for Demand Aggregation Exchanges

The exchange aggregates demands of all the buyers and also computes the maximum price it is willing to pay for the entire bundle. Table 2 shows the notation used in the formulation. The proposed iterative mechanism consists of multiple bidding rounds denoted by $t = 0, 1, 2, \ldots$ ($t = 0$ is the initial round). The aggregated bundle of all the buyers is therefore B_0. The exchange sets the value for $W(B_t)$, 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

Table 2. Notation for reverse Dutch combinatorial auction for demand aggregation exchanges

$t = 0, 1, 2, \ldots$	Iteration (or bidding round) number
B_t	Bundle to be procured in iteration t
B_0	Total aggregated bundle of all the buyers' demand
$W(B_t)$	Maximum willingness of the exchange to pay bundle B_t in iteration t
S_t	Bundle procured in round t
p_t	Average willingness of the exchange to pay for each item in round t
$V^*(S_t)$	Payment made by exchange for the subset S_t in iteration t. The payment is computed by solving GVA with reserve prices.
v_t	Average payment made for any item in iteration t
$\|S\|$	Cardinality of the set S
ϵ	Price increment per item in every iteration

bundles cannot be divided into price of individual items. Therefore we calculate p_t, the average willingness of the exchange to pay for each item in round t.

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

These average prices p_t are not used in the auction mechanism but are used to prove the bounds given in Section 4.3. The payment made by the exchange for the subset S_t in iteration t is $V^*(S_t)$. The average price paid by the exchange for each item procured is

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

We ignore the iterations in which no items are procured.

The exchange sets the reserve price of the seller for any bundle S in iteration t as $|S|v_{t-1}$. We use GVA with reserve prices [20] in each iteration to solve the allocation and payment problem efficiently. The bundle procured in round t is denoted by S_t. Therefore the integer programming formulation of the GVA problem with reserve prices in iteration t becomes

$$V^*(S_t) = \max \sum_{i \in M} \sum_{S \subseteq G} v(S,i)y(S,i) \tag{1}$$

$$\begin{aligned}
s.t.& \\
&\sum_{S \subseteq G} y(S,i) \leq 1 \;\; \forall i \in M \\
&\sum_{S \ni k} \sum_{i \in M} a(S,k)y(S,i) \geq q_k \;\; \forall k \;\; \forall S \subseteq G \\
&\sum_{i \in M} \sum_{S \subseteq G} v(S,i)y(S,j) \leq W(B_t) \;\; \forall S \subseteq M \\
&v(S,i)y(S,i) \geq |S|v_{t-1} \;\; \forall S \subseteq M, \forall j \in N
\end{aligned}$$

$$y(S,i) = 0,1 \;\; \forall S \subseteq G, \forall i \in M$$

4.2 Forward Combinatorial Dutch Auction for Supply Aggregation Exchanges

Here the exchange aggregates the supply from all the sellers and computes the total aggregated bundle and the minimum price at which it is willing to sell the aggregated bundle. Table 3 shows the notation for the formulation. In the

Table 3. Notation for the forward combinatorial Dutch auction for supply aggregation exchanges

$t = 0, 1, 2, \ldots$	Iteration (or bidding round) number
B_t	Bundle to be sold in iteration t
B_0	Total aggregated bundle available from of all the suppliers
$W(B_t)$	Minimum ask price of the exchange for pay bundle B_t in iteration t
S_t	Bundle sold in round t
p_t	Average ask price of the exchange for each item in round t
$V^*(S_t)$	Revenue earned by the exchange for the subset S_t in iteration t. The revenue is calculated by solving GVA with reserve prices.
v_t	Average revenue for any item in iteration t
ϵ	Price decrement per item in every iteration

combinatorial version of iterative forward Dutch auction ([11]), the seller starts with a high initial price and keeps on decreasing the price until the total bundle is sold. In the generalized Dutch mechanism for supply aggregation exchanges, the seller provides $W(B_t)$, total ask for the remaining bundle B_t to be sold in round t. We compute p_t, the average ask price of the exchange for each item in round t. The average ask prices p_t are not used in the auction mechanism but are used to prove the bounds given in Section 4.3. The payment earned by the exchange for the subset S_t in iteration t is $V^*(S_t)$. The average selling price for each item sold is

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

We ignore the iterations in which no items are sold i.e. $S_t \neq \phi \;\; \forall t$.

The reserve price for the remaining bundle S_t in iteration t is $W(B_t)$. Also the maximum payment to the buyers for any bundle S in iteration t is $|S|v_{t-1}$. Therefore the integer formulation of the GVA problem with reserve prices in iteration t becomes

$$V^*(S_t) = \min \sum_{j \in N} \sum_{S \subseteq G} v(S,j) y(S,j) \qquad (2)$$

$$s.t.$$

$$\sum_{S \subseteq G} y(S,j) \leq 1 \ \ \forall j \in N$$
$$\sum_{S \ni k} \sum_{j \in N} a(S,k) y(S,j) \leq q_k \ \ \forall k \ \ \forall S \subseteq G$$
$$v(S,j) y(S,j) \leq |S| v_{t-1} \ \ \forall S \subseteq G, \forall j \in N$$
$$\sum_{j \in N} \sum_{S \subseteq M} v(S,j) y(S,j) \geq W(B_t) \ \ \forall S \subseteq G$$
$$y(S,j) = 0,1 \ \ \forall S \subseteq G, \forall j \in N$$

4.3 Bounds for the Algorithms

The second stage problem in both demand aggregation and supply aggregation exchanges can be solved optimally in polynomial time. Therefore the bounds for these problems depend only on the bounds of the first stage problems. Thus bounds for these mechanisms are same as the bounds given for iterative Dutch combinatorial auctions in [11].

Upper Bound for Reverse Dutch Auction and Demand Aggregation Problem: The upper bound for the iterative reverse Dutch auction is $(1 + \frac{1}{2} + ... + \frac{1}{r})V(N)$, where

$$V(N) = \min \sum_{j \in N} \sum_{S \subseteq G} v_j(S) y(S,j), \text{and}$$

$$r = \max_{S \in K} |S|$$

Lower Bound for Forward Dutch Auction and Supply Aggregation Problem: The lower bound for the forward Dutch auction is $\frac{1}{l} V(N)$, where

$$V(N) = \max \sum_{i \in M} \sum_{S \subseteq G} v_i(S) y(S,i)$$

5 Numerical Experiments

We have derived the worst case (lower or upper) bounds for these iterative Dutch mechanisms. We have run our algorithms on some of the test cases suggested by various authors [21, 22, 2, 23]. We have used CPLEX^{TM} 8.0 to solve the various instances of GVA with reserve prices.

5.1 Experiments for a Demand Aggregation Exchange

We first conduct numerical experiments with demand aggregation exchanges by varying ϵ and the number of agents i.e. buyers and sellers. Since we solve the second stage problem exactly, the solutions vary according to the results of the

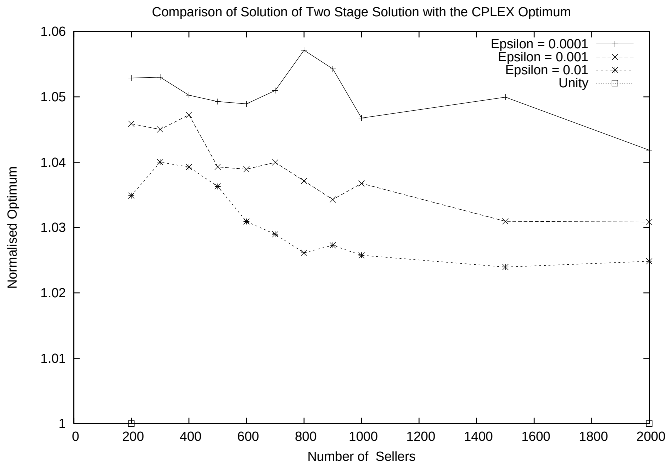

Fig. 1. Comparison of the demand aggregation results with the CPLEX optimum

first stage. The trends in the graphs are therefore very similar to those obtained in the reverse Dutch auctions discussed in [11].

The graph in Figure 1 shows that the solutions tend towards actual optimal solutions as ϵ tends towards unity. We also find the results of the approximate solutions are very good even when ϵ is very small. However, we see from the graph in Figure 2 that the time taken to solve the problems grows exponentially when ϵ is very close to 1. This happens because when ϵ is close to 1 the problem we solve in the first iteration is almost the as hard as original problem.

5.2 Experiments for a Supply Aggregation Exchange

Numerical experiments with supply aggregation exchanges by varying ϵ and the number of agents i.e. buyers and sellers are discussed in this section. In these experiments, we again solve the second stage problem exactly and therefore the solutions depend on the results of the first stage. The trends in the graphs are therefore very similar to those in the forward Dutch auctions discussed in [11].

The solutions tend towards the actual optimal solution as ϵ gets close to 1 (see Figure 3). But again the time taken grows exponentially when ϵ tends to 1 (see Figure 4).

6 Summary

In this paper, we have studied a useful class of combinatorial exchanges where we can have demand aggregation (for example, procurement exchanges) or supply aggregation (for example, exchanges selling excess inventory). We have shown that the allocation problem in these exchanges can be decomposed into two

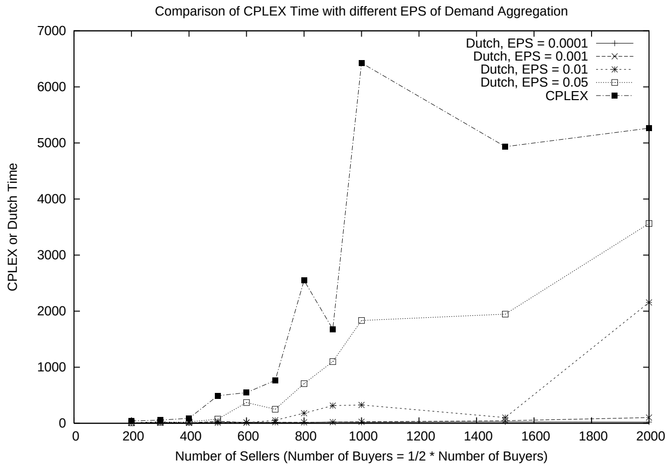

Fig. 2. Computation time comparison with CPLEX

stages: (a) Stage 1: Combinatorial auction (forward or reverse). (b) Stage 2: Assignment problem. Since the assignment problem in stage 2 can be solved exactly in polynomial time, we studied only the problem in the first stage. We have shown that the the iterative Dutch auction schemes proposed in [11] can be used for for solving the allocation problem in stage 1. This results in a computationally efficient way of solving aggregation exchanges to obtain near optimal allocations. We have conducted numerical experiments to show that the proposed approach and algorithms perform very well and give results which are very close to the optimal solutions.

The proposed approach can be used in all applications where aggregation combinatorial exchanges are relevant. The computational efficiency and the high quality of solutions will ensure that the mechanisms can be deployed in all practical applications. Procurement exchanges, excess inventory exchanges, and stock exchanges are a few examples.

In this paper, we have not considered game theoretic issues such as Vickrey payments, truthful bidding, etc. These are important issues that need to be addressed in future research.

References

1. Kalagnanam, J., Parkes, D.: Auctions, bidding, and exchange design. In Simchi-Levi, Wu, D., Shen, eds.: Supply Chain Analysis in the eBusiness Area. Kluwer Academic Publishers (2003)
2. de Vries, S., Vohra, R.: Combinatorial auctions, a survey. INFORMS Journal of Computing **15** (2003)
3. Hunsberger, L., Gross, B.J.: A combinatorial auction for collaborative planning. In: Proceedings International Conference on Multi-Agent Systems (ICMAS-00). (2000) 151–158

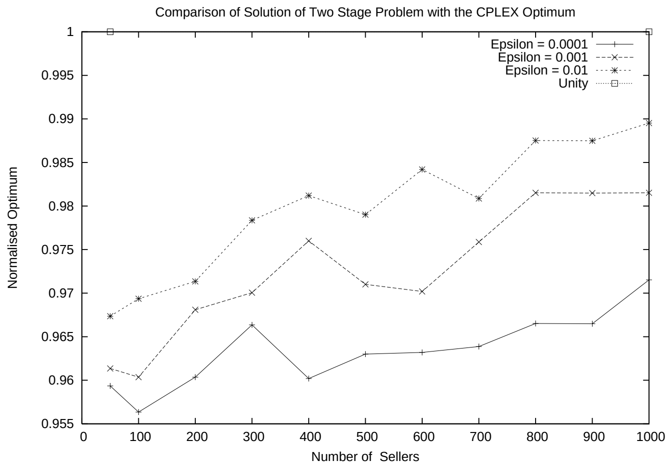

Fig. 3. Comparison of the supply aggregation results with the CPLEX optimum

4. Kameshwaran, S., Narahari, Y.: Innovative auction mechanisms for logistics marketplaces. In: Proceedings of the International Conference on Automation, Energy, and Information Technology, EAIT-2001, Indian Institute of Technology, Kharagpur. (2001)
5. Song, J., Regan, A.C.: Combinatorial auctions for transportation since procurement : The carrier perspective. Technical report, Institutee of Transportation Studies, University of California, Irvine (2002)
6. Ledyard, J.O., Olson, M., porter, D., Swanson, J.A., Torma, D.: The first use of a combined value auction for transprotation services. Technical Report 1093, California Institute of Technology (2000)
7. Eso, M., Kalagnanam, J., Ladanyi, L., Li, Y.: Winner determination in bandwidth exchanges. Technical report, RC22219 (W0110-087), IBM TJ Watson Research Center (2001)
8. Walsh, W.E., Wellman, M.P., Ygge, F.: Combinatorial auctions for supply chain formation. In: ACM Conference on Electronic Commerce. (2000) 260–269
9. Hohner, G., Rich, J., Ng, E., Reid, G., Davenport, A.J., Kalagnam, J.R., Lee, H.S., An, C.: Combinatorial and quantity discount procurement auctions provide benifits to mars and its suppliers. Technical report, IBM Research Report (2001)
10. Elmaghraby, W., Keskinocak, P.: Combinatorial auctions in procurement. Technical report, Dept. of Industrial and Systems Engineering, Georgia Institute of Technology (2002)
11. Biswas, S., Narahari, Y.: Iterative Dutch combinatorial auctions. Annals of Mathematics and Artificial Intelligence (2004) To appear in the special issue on the Foundations of Electronic Commerce.
12. Narahari, Y., Dayama, P.: Combinatorial auctions in electronic business. Technical report, Electronic Enterprises Laboratory, Department of Computer Science and Automation, Indian Institute of Science (2004)
13. Smith, T., Sandholm, T., Simmons, R.: Constructing and clearing combinatorial exchanges using preference elicitation. In: Proceedings of National Conference on Artificial Intelligence (AAAI-02). (2002)

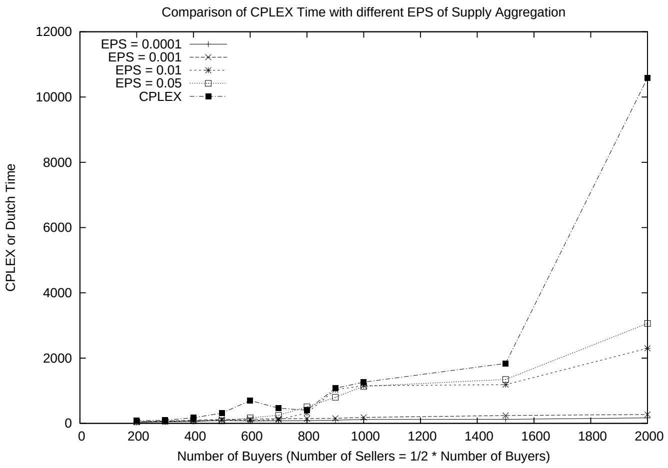

Fig. 4. Computation time comparison with CPLEX

14. Dayama, P., Narahari, Y.: Design of multi-unit electronic exchanges through decomposition. Technical report, Electronic Enterprises Laboratory, Department of Computer Science and Automation, Indian Institute of Science (2003)
15. Kothari, A., Sandholm, T., Suri, S.: Solving combinatorial exchanges: Optimality via a few partial bids. In: AAAI. (2002)
16. Parkes, D.: Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency. PhD thesis, Department of Computer and Information Science, University of Pennsylvania (2001)
17. Parkes, D.C., Kalagnanam, J., Eso, M.: Achieving budget-balance with Vickrey-based payment schemes in exchanges. In: 17th International Joint Conference on Artificial Intelligence (IJCAI-01). (2001)
18. Biswas, S.: Iterative Mechanisms for Combinatorial Auctions and Exchanges. PhD thesis, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India (2004)
19. Bikchandani, S., Ostroy, J.M.: The package assignment model. Technical report, Anderson Graduate School of Management and Department of Economics, UCLA (2001)
20. Ausubel, L., Cramton, P.: Vickrey auctions with reserve pricing. Technical report, Working Paper, University of Maryland (1999)
21. Sandholm, T.: An algorithm for optimal winner determination in combinatorial auctions. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-99). (1999) 542–547
22. Boutilier, C., Goldszmidt, M., B., S.: Sequential auctions for the allocation of resources with complementarities. In: International Joint Conferences on Artificial Intelligence. (1999)
23. Kevin Leyton-Brown, Mark Pearson, Y.S.: Towards a universal test suite for combinatorial auction algorithm. In: ACM Conference on Electronic Commerce, 2000. (2000)