Design and Evaluation of a Combinatorial Double Auction for Resource Allocations in Grids*

Li Li[1], Yuanan Liu[1], David Hausheer[2], and Burkhard Stiller[2,3]

[1] School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, P. R. China
[2] Communication Systems Group CSG, Department of Informatics IFI, University of Zürich, Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
[3] Computer Engineering and Networks Laboratory TIK, ETH Zürich, Gloriastrasse 35, CH-8092 Zürich, Switzerland
lily82.bupt@gmail.com, yuliu@bupt.edu.cn, hausheer@ifi.uzh.ch, stiller@ifi.uzh.ch

Abstract. Offering Grid services in an open market determines an optimization case for finding the best suitable resource allocation for a given number of requests and existing resources. Thus, appropriate resource allocation schemes, supporting accounting, are required in addition to a pricing scheme, which supports financial fairness criteria. The newly developed Resource Allocation Model for the Combinatorial Double Auction (RAMCoDA) achieves these requirements, while being incentive compatible.

1 Introduction and Related Work

Commercial Grid services need to support resource allocations, which should be incentive-compatible. Therefore, resource allocations in Grids need the support of a suitable accounting and pricing (often termed billing) scheme. As economic theory tells, auctions do have the potential, if applied in a sensible manner, to achieve the incentive compatibility. With respect to the Grid services market, Combinatorial Auctions (CA) can represent satisfying characteristics. Within CAs, the user can bid for combinations of resources, on which tasks can be executed, while improving economic efficiency and maximizing the revenue of the Grid. However, existing CA-based resource allocations [1], [6] focus on the users' side only and they do not define any specific mechanisms for pricing and accounting resources. Since the CA's pricing algorithm is a computationally complex task, existing work regard the final price paid by the agent often as its bid or the simple division of the auctioneer's total income. This may lead to a misrepresentation of the agents' true valuations, which cannot ensure incentive compatibility.

* This work was supported partly by the European Union through the EC-GIN project (STREP) under Grant No. FP6-2006-IST-045256. The authors acknowledge fruitful discussions with all their project partners.

P. Reichl, B. Stiller, and B. Tuffin (Eds.): ICQT 2009, LNCS 5539, pp. 36–47, 2009.

Thus, a novel resource allocation model based on Combinatorial Double Auctions (CoDA) is proposed, which is suitable for accounting and pricing purposes. The CoDA combines both advantages of a Double Auction (DA) and a CA. Analytical experiments of the newly developed Resource Allocation Model for the Combinatorial Double Auction (RAMCoDA) show that the new scheme is effective and incentive-compatible. Existing pricing schemes in related work can be classified into three main categories: Bargaining Models [2], Commodity Market Models [5], [10], and Auction Models [6], [4], [9]. Auction Models include either one-to-many or many-to-many interactions. The DA is widely used for many-to-many auctions, in which buyers and sellers are treated symmetrically with buyers and sellers submitting bids. Instead of selling items of resources individually, in a CA the seller allows bids on bundles of items, enabling bidders to deal with entities of direct interest and avoiding the risk of obtaining incomplete bundles. The auction based pricing schemes include an autonomous pricing mechanism as proposed by [9], in which prices are decided by Grid traders within the trading process, a type of autonomous pricing mechanism proposed by [4], in which consumers and producers are autonomous agents that make their own decisions according to their capabilities and their local knowledge, and another pricing strategy for combinatorial Grids introduced in [6], where resource agents administrate available memory, CPU, network bandwidth, and disk capacities on the supply side. Further details on related auctions and pricing schemes can be obtained in [3].

The distinguishing characteristic of RAMCoDA as herein proposed is determined by the use of CoDA onto resource allocation and pricing in Grids. The newly designed model enables to obtain the complete allocation information and trade price as described in the following sections.

While Section 2 introduces the underlying resource allocation model of RAMCoDA, Section 3 defines the corresponding pricing algorithm. Both are simulated in Section 4 and a summary is given in Section 5.

2 CoDA-Based Resource Allocation Model

For the novel CoDA-based resource allocation model (cf. Fig. 1.) each Grid user operates a User Broker (UB). Within the UB, the Resource Discovery component is responsible for finding resources according to users' requirements; it contacts the Grid Information Service (GIS) to obtain the list of resources that matches these requirements. The Auction Agent in UB is responsible for generating resource combinations based on the list of resources returned by the GIS. For each combination of resources, it generates a bid within the user's budget and submits the bid and the corresponding combination to the Grid Market Auctioneer (GMA). The Price Depository component is responsible for storing price information related to the task. At a later stage, this information can be used for accounting. The Job Management Agent is responsible for sending the user job to resources and collecting the results.

The GIS provides resource registration services and maintains a list of resources available in the Grid. The GIS may be implemented in a decentralized

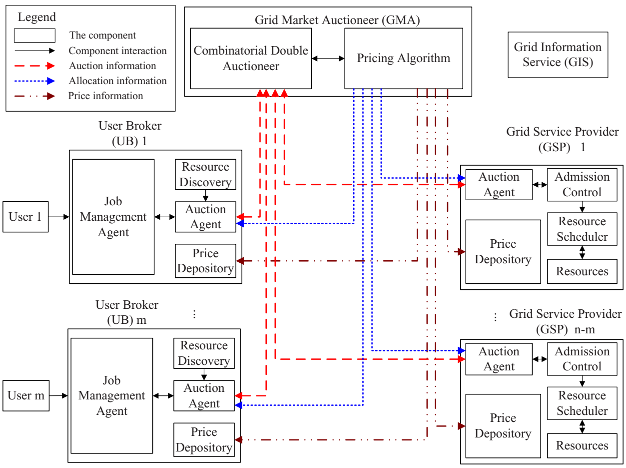

Fig. 1. Resource Allocation Model for the Combinatorial Double Auction (RAMCoDA)

manner, *e.g.*, based on a Distributed Hash Table (DHT). Each Grid Service Provider (GSP) will register all resources it can provide at the GIS. The Auction Agent in GSP is responsible for generating the resource combination it would sell. For each resource combination, it generates a bid and submits it together with the corresponding combination description to the GMA. The Admission Control component receives the auction result from the Auction Agent and decides, whether the tasks sent from the UB will be managed or not. The Resource Scheduler is responsible for allocating all tasks to the corresponding resources. Within the GMA, the Combinatorial Double Auctioneer is responsible for collecting the combination of resources and corresponding bids sent by UBs and GSPs. Based on the that information it runs the CoDA algorithm to determine the winner UBs and GSPs. Additionally, it sends the result obtained to UBs and GSPs. Finally, the Pricing Algorithm is responsible for generating particular allocation results and corresponding price information. The price calculated will be sent to all UBs and GSPs, who did participate in the trade.

RAMCoDA mainly differs from the allocation model in [1] in terms of the applied auction model and the pricing algorithm (cf. Section 3) in the GMA and the Price Depository components in the UB and GSP. Consequently, interactions between the components are different. In [1], only UBs submit their bids, while in RAMCoDA both UBs and GSPs submit their bids and offers, respectively.

3 The Pricing Algorithm

The pricing algorithm is a key part of any DA approach. Thus, for the resource allocation the algorithm proposed follows five steps to determine the best suited resource price.

1. Assuming that there are n participants in a CoDA, including m users and $n-m$ GSPs, UBs and GSPs submit resource combinations (bundles) and bids, in form of B_j, to the GMA. B_j can be specified as (a_j, p_j), where $a = (a_{1j}, ..., a_{ij}, ...a_{kj})$, a_{ij} is the quantity of item i requested ($a_{ij} > 0$) or supplied ($a_{ij} < 0$) in the bundle j. Here suppose a UB/GSP is allowed to ask for one bundle in the auction, so the subscript j could identify both a bundle and a UB/GSP. The symbol k denotes that there are k resource items to be considered jointly in the auction. K is the set of the items. p_j is the amount the bidder is willing to pay for the bundle j, if $p_j > 0$, it is regarded as a buying price; otherwise it is regarded as a selling price. The combinatorial double auctioneer runs the CoDA algorithm represented in (1) and receives the result x_j [8]. The objective of CoDA is to maximize the total trade surplus, while satisfying the constraint that the number of units selected by buying bundles does not exceed the number provided by the selected selling bundles for each item.

$$max \sum_{j=1}^{n} p_j x_j, \quad s.t. \sum_{j=1}^{n} a_{ij} x_j \le 0 \quad \forall i \in K, \tag{1}$$
$$with \ x_j \in \{0,1\} \ \forall j \in \{1, ..., n\}.$$

Afterwards, the GMA informs UBs and GSPs about the bid's acceptance or rejection and requests the winner GSPs to reserve the resources awarded. Loser participants can renew their bids in the next round. The winner participants are denoted as the traders in the following.

2. For each trader, the average bid is calculated as

$$p_j^a = \frac{p_j}{\sum_{i=1}^{k} a_{ij}}. \tag{2}$$

All traders are buyers or sellers. Buyers are ranked in the decreasing order of the average bid; the result is denoted as the buyer list b^l. Accordingly, sellers are ranked in the increasing order of the average bid; the result is the seller list s^l. All sellers are classified by the resource item. The algorithm achieves s_i^l, $i \in \{1, ..., k\}$ for each item of the resource, while q_i^l represents the resource quantity list corresponding to the s_i^l.

3. Generating the average trade price matrix p^m, in which $p^m(s,t)$ represents the average trade price, when the sth buyer in the list b^l trades with the tth seller in the list s^l, the value is calculated as follows in (3):

$$p^m(s,t) = \frac{p_{b^l(s)}^a + p_{s^l(t)}^a}{2}. \tag{3}$$

4. The resource allocation and pricing is done in order from the first buyer in b^l to the last one. Buyers will be matched with the first seller holding the required resources in the s^l for the allocation. If requirements of this buyer can be satisfied, the algorithm will update results and go to the next

buyer, otherwise, the buyer will be matched with the next seller for surplus requirements. Moreover, R_i^a and R_i^p, which are both $g \times h$ matrices, represent particular allocation and pricing results of the resource i, respectively. Here g denotes the number of buyers in b^l, while h denotes the number of sellers in s^l. The algorithm is defined as follows:

Input:$B, b^l, s^l, s_i^l, q_i^l$ Output:$R_i^p, R_i^a; i \in \{1, ..., k\}$

(a) Initialization:
$s = 1$, $i = 1$, $R_i^p = [0]_{g\times h}$, $R_i^a = [0]_{g\times h}$.

(b) Inquire the average trade price matrix p^m; compare the buyer's requirement $a_{ib^l(s)}$ with $q_i^l(m)$
$m \leftarrow$ the location of the first non-zero quantity in the list q_i^l
$t \leftarrow$ the location of seller $s_i^l(m)$ in the list s^l
If $q_i^l(m) \geq a_{ib^l(s)}$
  $R_i^p(s,t) = R_i^p(s,t) + a_{ib^l(s)} \cdot p^m(s,t)$;
  $R_i^a(s,t) = R_i^a(s,t) + a_{ib^l(s)}$;
  $q_i^l(m) = q_i^l(m) - a_{ib^l(s)}$;
  $a_{ib^l(s)} = 0$;
  Go to Step (c);
else
  $R_i^p(s,t) = R_i^p(s,t) + q_i^l(m) \cdot p^m(s,t)$;
  $R_i^a(s,t) = R_i^a(s,t) + q_i^l(m)$;
  $a_{ib^l(s)} = a_{ib^l(s)} - q_i^l(m)$;
  $q_i^l(m) = 0$;
  Repeat Step (b);

(c) Store the temporary allocation result R_i^a and pricing information R_i^p, determine whether all the resource requirements of the *sth* buyer are satisfied,
If all requirements are satisfied
  Go to Step (d);
else
  $i = i + 1$;
  Go to Step (b);

(d) Determine whether all buyers' requirements are satisfied, i.e., whether get to the end of the b^l,
If end of the b^l
  EXIT;
else
  $s = s + 1$;
  $i = 1$;
  update R_i^p, R_i^a, q_i^l;
  Go to Step (b);

Finally, each trader will receive a trade price represented by the vector p^b or p^s:

$$p^b = (\sum_{t=1}^{h} \sum_{i=1}^{k} [R_i^p]_{g\times h})^T,$$
$$p^s = \sum_{s=1}^{g} \sum_{i=1}^{k} [R_i^p]_{g\times h}, \tag{4}$$

where T denotes the transposition of the matrix.

5. Finally, the GMA sends the related information in those vectors p^b, p^s, and R_i^a, R_i^p to each trader.

4 Simulation Results and Discussion

Based on the model and pricing algorithm presented above, a performance investigation has been undertaken. While the functionality of RAMCoDA shows the interactions needed to obtain the result of a corresponding price, the performance needs an analytical approach. Thus, a simulation has been performed. Experiments were run in Matlab on a Pentium D dual-core CPU 2.8 GHz with 1 GB memory. Key rules for the parameter selection are as follows: for buyers, the value of each resource item is within the range $[l_i^b, u_i^b]$, for sellers, the value of each resource item is within the range $[l_i^s, u_i^s]$, with the constraint $l_i^s \leq l_i^b$, $u_i^s \leq u_i^b$ to lower the possibility that too many bids of sellers are higher than bids of buyers. Here $l_i^b, u_i^b, l_i^s, u_i^s$ are all the fixed parameters of the simulation. The demand and supply quantity of resource i from each participant, i.e.$|a_{ij}|$, is uniformly distributed over the interval $[0, d_i]$ and $[0, s_i]$, respectively. Based on the assumption above, each buyer can give its valuation of his resource combination within the range $\left[\sum_{i=1}^{k} a_{ij} l_i^b, \sum_{i=1}^{k} a_{ij} u_i^b\right]$, while seller's valuation is within $\left[\sum_{i=1}^{k} a_{ij} u_i^s, \sum_{i=1}^{k} a_{ij} l_i^s\right]$. In the following simulation, all participants bid for resource combinations according to their true valuation, which means bids are equal to those valuations.

The particular simulation undertaken considers 3 items of resources in the Grid, denoted as A, B, and C. These resources may include in a real-world scenario storage, access bandwidth, and a certain software library needed to run the Grid's task of a weather simulation. In the case considered, 20 participants are involved in the CoDA, including 10 UBs and 10 GSPs. For UBs, the value ranges of 3 items of resources are [5, 10], [10, 15], [15, 20] respectively, while for GSPs, the ranges are [4, 8], [8, 12], [12, 16] respectively. Moreover, $d_i = s_i = 20$ has been selected. The parameter settings can be found in Tab.1. Through solving the CoDA represented in (1), participants 1, 2, 3, 5, 6, 7, 8, 9, 10 and 11, 12, 13, 14, 16, 17, 19, 20 are the winning bidders. The bids of 4, 15 and 18 are rejected in this round.

The allocation and pricing results are shown in Fig. 2. Buyer list b^l has the same order as the row order of R_i^a and R_i^p, while the seller list s^l has the same order as the column order of these matrices. It can be seen that Fig. 2 gives the complete information of the allocation and all trade prices. Take buyer 3 for example: his demand is (0 2 7) – determining the request for 2 units for resource B and 7 units for resource C, respectively – and the bid is 147. His trade information can be obtained from the corresponding row based on his position in b^l, i.e. the first row, in the R_i^a and R_i^p matrices. As marked in R_i^a in Fig. 2, all requirements of buyer 3 are satisfied by 3 sellers jointly, thus, here the first seller in s^l, i.e. 17, provides 2 units of resource B and 2 units of resource C; the second seller in s^l, i.e. 16, provides 2 units of resource C; and the third seller in s^l, i.e. 14, provides 3 units of resource C. They will charge this buyer 23, 23, 24.8 and 38.3 monetary units, respectively, as marked in R_i^p in Fig. 2. Similarly, the trade information of each seller can be gained from the corresponding column

Table 1. Parameter settings for each participant

No.	A	B	C	p	No.	A	B	C	p
1	3	10	4	253	11	-13	-20	-11	-461
2	6	16	8	417	12	-12	-1	-19	-326
3	0	2	7	147	13	-7	-15	-3	-250
4	5	13	14	399	14	-7	-20	-16	-396
5	11	14	20	683	15	-11	-12	-7	-336
6	8	9	16	509	16	-19	-15	-2	-303
7	15	9	9	392	17	-14	-10	-2	-174
8	16	6	1	200	18	-10	-1	-20	-366
9	15	16	6	442	19	-7	-14	-13	-349
10	13	19	1	431	20	-13	-9	-6	-272

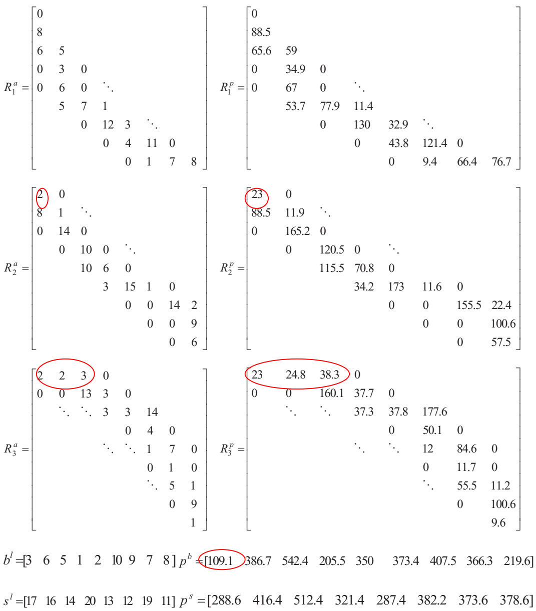

Fig. 2. Selected Results of the Allocation and Pricing Simulation

in the matrices. The total payment of each buyer and the total income of each seller are shown in p^b and p^s. As marked, the total payment of buyer 3 is 109.1.

It can be seen that RAMCoDA is very suitable for resource allocation in Grid. This is due to the following reasons: (a)comparing with the allocation based on the single-item auction, which needs multiple rounds of auctions for a single item of resource to be auctioned off, RAMCoDA performs more efficiently, *e.g.*, the proposed pricing algorithm determines the resource allocation and pricing of 17 traders in a single round in the above simulation; (b)the combination of both advantages of the DA and the CA, including the prevention of a monopoly, while considering requirements of both sides and permitting bidding for the resource combination requested by the task; and (c)it can achieve the explicit allocation and trade price information as well as the incentive compatibility characteristic, which are missing in a pure CA-based approach.

The achieved incentive compatibility characteristic is explained in the following. Due to the assumption that all participants will bid for resource combinations according to their true valuations, for buyers it holds that *utility = true valuation - trade price = bid - trade price*, while for sellers it holds that *utility = trade price - true valuation = trade price - bid* [7]. If there is a buyer who attempts to obtain a better utility through misrepresenting the true valuation, the results can be classified as follows: If $bid < true\,valuation$, then he might be (1) rejected by the CoDA algorithm because his bid is too low, then his utility becomes 0; (2) still one of the winner bidders, but due to the reason that this bid change will lead to the change of his location within list *bl*, this causes him to trade with the higher bid sellers and finally obtain the lower utility; (3) still one of the winner bidders, and can get the lower final trade price because of his lower bid, even though he has to trade with the higher bid sellers. If $bid > truevaluation$, he will generally obtain the lower utility except the situation in which he has the chance to trade with an extremely low bid seller. Few participants would like to act in this way.

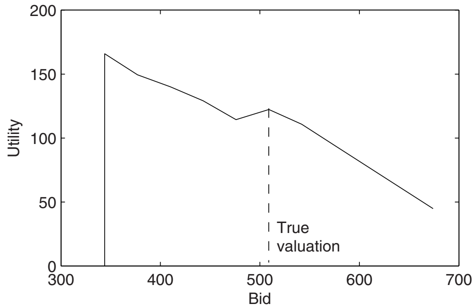

Fig. 3. The utility as a function of the bid

Fig. 3 takes buyer 6 in the previous experiment for example, and shows the utility as a function of his bid. Here assuming all other participants still hold their original bids as shown in Tab.1. It can be seen from Fig. 3, when buyer 6 is under-reporting the true valuation, the fluctuation in the figure is corresponding to the three situations discussed above. He can only get the higher utility under situation (3); however, because there is still the risk of getting a lower utility or even being rejected by the CoDA, and the buyer has no idea about other participants' bids, it will be difficult for him to decide how much to under-report, and arbitrarily under-reporting may actually lower his utility, as shown in Fig. 3. The situation is similar for the sellers. Therefore, bidding the true valuation is still the optimal strategy for each participant.

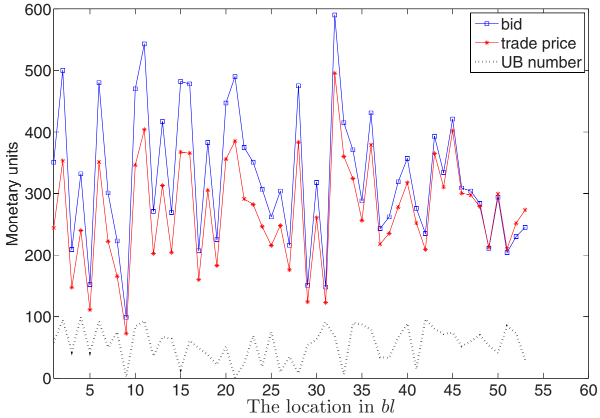

Fig. 4. Results of the Pricing Algorithm (for buyers,n=200)

In order to demonstrate the performance of the proposed pricing algorithm, a resource allocation on a larger scale has been simulated. It is assumed that 200 participants are involved in the CoDA, including 100 UBs and 100 GSPs. For all participants, the value ranges of 3 items of resources are [5, 20], [10, 25], [15, 30] respectively, and $d_i = s_i = 10$ is selected. Since this selection has been done randomly and through solving the CoDA, 53 UBs and 54 GSPs are the winning bidders. In any other case of randomly selected parameter values, different UBs and GSPs would be winning, however, the algorithm proposed would work the same way. Therefore, these results of the algorithm are illustrated in Fig. 4 and Fig. 5, representing a demonstration case. It presents the comparison between the original bid and the trade price. The horizontal axes in Fig. 4 and Fig. 5 represent the corresponding position in b^l and s^l, respectively. Dotted lines in the figure do not relate to the price, they determine the original UB or GSP number corresponding to the number on the horizontal axis.

For example, 21 on the horizontal axis in Fig. 4 denotes the 21th buyer in b^l; from the dotted line, it can be seen that it represents the first (no. 1) UB in the simulation. The original bid and trade price of this UB are 490 and 385.1 monetary units, respectively. It can be seen that the algorithm proposed can complete the resource pricing efficiently, the trend of the trade price obtained is similar to that of the original bid. Moreover, the buyers (sellers) in the front of the list $b^l(s^l)$ can receive a compensation, while the last set of buyers (sellers) will pay (get) the higher (lower) price for the service than their original valuations, i.e., they will get negative utilities. If the trader only enters into a transaction under a positive utility, the trader with the negative utility can quit the current resource allocation, and shall renew its bid in the next round.

Fig. 6 shows the efficiency of RAMCoDA. Three kinds of parameter settings are considered here. For parameter setting 1, the rules of these parameter value

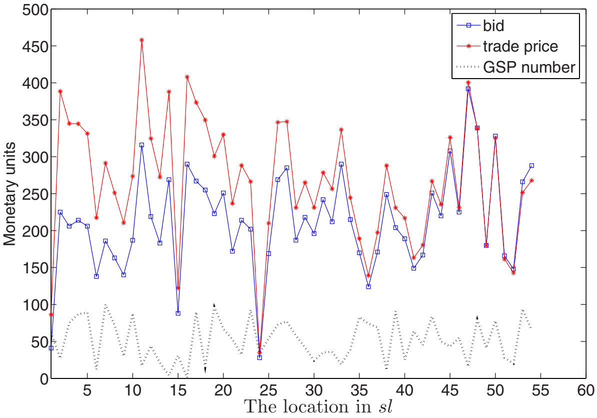

Fig. 5. Results of the Pricing Algorithm (for sellers,n=200)

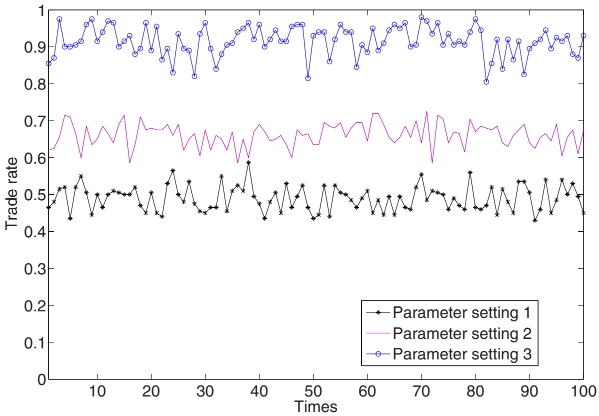

Fig. 6. The trade rate for 3 kinds of parameter settings(100 simulation rounds)

selections are the same as those in Fig. 4 and Fig. 5. For parameter setting 2 and 3, the resource value ranges of UBs change to [10, 25], [15 30], [20, 35] and [20, 30], [25, 35], [30, 40]. Trade rate is defined as the number of final traders divided by the number of the original participants. It can be seen that RAMCoDA can obtain a high trade rate in a one round auction, and the trade rate will increase remarkably when the resource value ranges of UBs is higher than those of GSPs. Since the bids of the UBs are generally higher than those of the GSPs, it is possible for an auction to get more matches between the demand and supply.

Fig. 7 gives the average trade price level for the buyers and sellers for 100 simulation rounds. All the parameters follow the rules of parameter setting 1. The average trade price level is defined as the mean of all the buyers' or sellers' average trade prices. While for each trader, the average trade price equals to

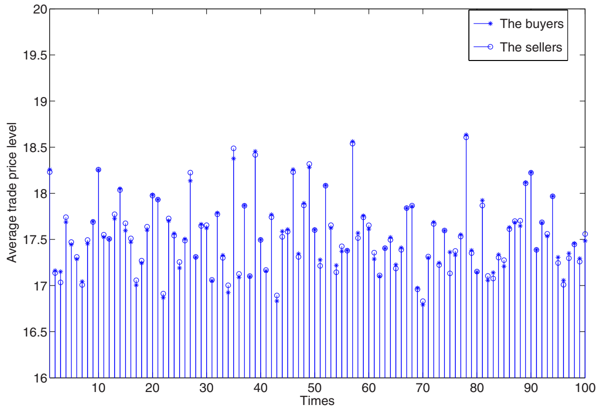

Fig. 7. The average trade price level for the buyers and sellers

the trade price of this trader divided by his total demand or supply. It can be noticed in Fig. 7 that the average trade price levels for the buyers and sellers are similar for a certain time, and in 100 simulation rounds, the fluctuation range of the level is less than 2, which can be regarded as a stable price level since the fluctuation range is less than 13.3% of the average price level.

5 Summary and Conclusions

The newly developed RAMCoDA determines an effective means in support of Grid services resource allocation, accounting, and pricing. RAMCoDA's pricing algorithm is incentive compatible. Analytical investigations presented that the allocation and pricing results need a single round of auctions only, while all requirements of both users and GSPs are taken into account. Finally, the algorithm achieves an explicit allocation and trade price, a high trade rate and a stable price level.

Therefore, RAMCoDA offers a valuable approach for Grid service providers within a commercial situation to market their services. The scheme's simplicity and effectiveness determine reasonable arguments for a practical approach. In addition, the resource usage obtained is fair and can be applied to accounting purposes, thus, GSPs and users will benefit from the approach proposed.

References

1. Das, A., Grosu, D.: Combinatorial auction-based protocols for resource allocation in Grids. In: 19th IEEE International Parallel and Distributed Processing Symposium, Colorado, U.S.A., pp. 23–30 (2005)
2. Ghosh, P., Roy, N., Das, S.K., Basu, K.: A pricing strategy for job allocation in mobile Grids using a non-cooperative bargaining theory framework. Journal of Parallel and Distributed Computing 65(11), 1366–1383 (2005)

3. Li, L., Liu, Y.A., Stiller, B.: Combinatorial double auction-based scheme for resource allocation in Grids. University of Zürich, Department of Informatics, CSG, Technical Report No. 2008.05, Zürich, Switzerland (2008)
4. Pourebrahimi, B., Bertels, K., Kandru, G.M., Vassiliadis, S.: Market-based resource allocation in Grids. In: Second IEEE international Conference on e-Science and Grid Computing, Amsterdam, The Netherlands, pp. 80–88 (2006)
5. Stuer, G., Vanmechelen, K., Broeckhove, J.: A commodity market algorithm for pricing substitutable Grid resources. Future Generation Computer Systems 23(5), 688–701 (2007)
6. Schwind, M., Gujo, O., Stockheim, T.: Dynamic resource prices in a combinatorial Grid system. In: 8th IEEE International Conference on E-Commerce Technology and 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, San Francisco, California, U.S.A., pp. 49–54 (2006)
7. Xia, M., Koehler, G.J., Whinston, A.B.: Pricing combinatorial auctions. European Journal of Operational Research 154(1), 251–270 (2004)
8. Xia, M., Stallaert, J., Whinston, A.B.: Solving the combinatorial double auction problem. European Journal of Operational Research 164(1), 239–251 (2005)
9. Yang, J., Yang, S.B., Li, M.S., Fu, Q.F.: An autonomous pricing strategy toward market economy in computational Grids. In: International Conference on Information Technology: Coding and Computing, Las Vegas, Nevada, U.S.A., pp. 793–794 (2005)
10. Zhao, X.G., Xu, L.T., Wang, B.: A dynamic price model with demand prediction and task classification. In: The Sixth International Conference on Grid and Cooperative Computing, Urumchi, P.R. China, pp. 775–782 (2007)