

Maximal Matching for Double Auction*

Dengji Zhao^{1,2}, Dongmo Zhang¹, Md Khan¹, and Laurent Perrussel²

¹ Intelligent Systems Laboratory,
University of Western Sydney, Australia
{dzhao,dongmo,md.khan}@scm.uws.edu.au

² IRIT, University of Toulouse, France
laurent.perrussel@univ-tlse1.fr

Abstract. We study the problem of mechanism design for a double auction market where multiple buyers and sellers buy and sell a commodity. We design and implement a matching algorithm that maximizes market liquidity, including the number of transactions and buy/sell-volume. We prove that, given the number of matches, the algorithm also maximizes auctioneer's profit. Based on the CAT Tournament (Trading Agent Competition Market Design) platform, we show with experiments that the new matching method not only increases market liquidity but also significantly improves market share and auctioneer's profit in the long term, compared with equilibrium matching, the most commonly used matching method.

1 Introduction

A double auction is a market mechanism that allows multiple buyers and sellers trade simultaneously [1,2,3]. Given the supply and demand of sellers and buyers (i.e. traders), a double auction is characterized by (i) how to match bids (offers to buy) and asks (offers to sell), and (ii) what price to execute each matched ask-bid pair (clearing price). Similar to the design of other market mechanisms, the main concerns of double auction design include *incentive compatibility* (IC), *market liquidity*, *market efficiency* (social welfare) and *market profit* (auctioneer's revenue). Although all these properties are desirable, it is impossible for a double auction mechanism to possess all of them. McAfee and Wurman *et al.* showed that there is no double auction mechanism that is both efficient and incentive compatible [4,5].

As a tradition in the research of mechanism design, most existing work on double auctions put emphasis on incentive compatibility and social welfare (e.g. [4,5,3]). However, the other properties, especially market liquidity, are equally important. The liquidity of a market, which indicates the number of transactions and buy/sell volume of the market, not only effects market profit and social welfare but also flags the success and reputation of the market. A market with high liquidity can attract traders to the market as it brings trading opportunities.

* This research was supported by the Australian Research Council through Discovery Project DP0988750.

This paper makes an endeavour at the design of non-IC double auction mechanism. Without restriction of incentive compatibility, we can maximize matches and set clearing prices in terms of bidding prices, therefore it becomes possible to maximise market liquidity and optimize other market indicators, such as market share, market profit and social welfare. To this end, we propose a new matching algorithm, named *maximal matching* (MM), for any double auction mechanism and prove that it maximises the matches of incoming bids and asks. By comparing the algorithm with the most commonly used matching method, *equilibrium matching* (EM), we demonstrate through a set of experiments that maximal matching not only maximizes market liquidity, but also significantly improves market share (in terms of the number of traders) and auctioneer's profit.

The rest of the paper is organized as follows. Section 2 introduces the market model and lists the commonly used criteria for double auction mechanism design. Section 3 presents maximal matching algorithm. Section 4 analyses the properties of maximal matching and compares maximal matching with equilibrium matching. Section 5 shows the experimental results of both maximal and equilibrium matching, which gives a support to the theoretical results presented in Sect. 4. Finally we conclude the work.

2 Preliminaries

2.1 The Market Model

We study the problem of matching in a double auction market where multiple buyers and sellers buy and sell one commodity. Let $T = S \cup B$ be a set of traders, where S is the set of sellers, B is the set of buyers, and $S \cap B = \emptyset$ ¹. A *shout* is a message that a trader sends to the auctioneer (the double auction market) for either buying or selling one unit of the commodity with a specified price². Let Ω be the set of all possible shouts. For each shout $s \in \Omega$, we write $p(s)$ to represent the bidding price of s (non-negative), and $t(s)$ to denote the trader who sends s .

A double auction is running as follows: each trader submits a number of shouts to the auctioneer, and the auctioneer decides which sell and buy shouts to be matched and what price to execute each match.

Definition 1. An *ask* is a shout $a \in \Omega$ such that $t(a) \in S$. A *bid* is a shout $b \in \Omega$ such that $t(b) \in B$. For any finite set $X \subset \Omega$, we let $X^{ask} = \{x \in X : t(x) \in S\}$ and $X^{bid} = \{x \in X : t(x) \in B\}$. A *matching* of X , denoted by $M(X)$, is a collection of pairs $\{(a_1, b_1), (a_2, b_2), \dots\}$, where $a_i \in X^{ask}$, $b_i \in X^{bid}$, $p(a_i) \leq p(b_i)$, and $a_i \neq a_j$, $b_i \neq b_j$ when $i \neq j$.

¹ In the real world, a trader can be both a seller and a buyer for the same commodity. In such a case, we model it as two different roles as the decision making for selling and buying is different.

² For sake of simplicity, we assume that each shout contains only one unit of the commodity. It is possible to extend the model to handle a shout with multiple units by splitting the shout into many shouts with one unit each.

2.2 Objectives Considered in Double Auction Mechanism Design

The following desirable objectives or desiderata are mostly considered in double auction design [6]:

- *Incentive Compatibility.* A double auction is said to be incentive compatible if all of the participants maximize their utilities when they truthfully reveal any private information asked for by the auction.
- *Liquidity Maximization.* The goal is to maximize: (a) number of transactions, (b) sell volume: the total amount of cleared asks, and (c) buy volume: the total amount of cleared bids [3].
- *Profit Maximization.* Each pair of ask and bid that are matched produces a profit, which is the difference between the bid price and the ask price. This objective is to maximize the sum of these differences, over all matched pairs.
- *Social Welfare Maximization (Efficiency).* This objective corresponds to maximizing the goods of the buyers and sellers in aggregate. That is, the goods are allocated to the agents who value them most highly.
- *Individual Rationality.* A double auction is individual rational if it gives its traders non-negative utility/profit.

3 Matching Algorithm

This section will present our maximal matching algorithm. Before doing that, let us briefly introduce the most commonly used matching policy for double auction markets—*equilibrium matching* (EM).

3.1 Equilibrium Matching

Equilibrium Matching is used to find a *uniform price* p^* (equilibrium price) which balances the bids and the asks going to be matched so that all the bids with price $p \geq p^*$ and all the asks with price $p \leq p^*$ are matched [1]. EM can be easily implemented as follows:

1. Sort all asks (bids) in ascending (descending) order w.r.t. their price.
2. Based on this sort order, starting at the top, add each ask-bid pair to the result matching, if ask's price is less than or equal to bid's price.

Algorithm 3.1 describes the above implementation. A more deliberate, and yet popular, implementation, named *4-Heap*, is given by Wurman *et al.* [5]. A uniform price is normally determined by the last matchable or the first unmatchable shout pair w.r.t. the matching order in Algorithm 3.1 [4,5].

3.2 Maximal Matching

A double auction with EM can be incentive compatible or efficient (but not both) with some special pricing policies [4,5]. However, no double auction mechanism with EM can maximize liquidity as the uniform clearing price might prohibit some matchable shouts from being matched. In order to maximise the number

Algorithm 3.1. EquilibriumMatching

```

Input: Asks: sorted in ascending order, Bids: sorted in descending order
Output: Matching
1 begin
2   Matching  $\leftarrow \emptyset$ ;  $I \leftarrow 1$ ;
3   while Asks  $\neq \emptyset$  and Bids  $\neq \emptyset$  do
4     Ask  $\leftarrow$  read I-th ask from Asks; Bid  $\leftarrow$  read I-th bid from Bids;
5     if  $p(\textit{Ask}) \leq p(\textit{Bid})$  then
6       Matching  $\leftarrow$  Matching  $\cup \{(Ask, Bid)\}$ ;  $I \leftarrow I + 1$ ;
7     else
8       jump out while loop;
9     end
10  end
11 end

```

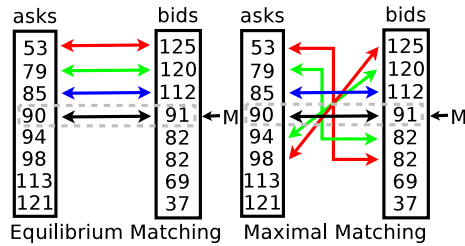


Fig. 1. Equilibrium Matching vs. Maximal Matching

of matches/transactions, it is essential to allow different matches cleared at different prices³. Otherwise, some matches might be cleared at a price which is not between the ask price and the bid price of them, i.e. it will act against individual rationality, which is a basic assumption of double auction mechanism design. Based on this idea, we introduce a new matching algorithm, named *Maximal Matching* (MM), as we shall prove that it maximises the number of matches in Sect. 4. The algorithm is given in Algorithm 3.2, which can be summarised in the following five steps:

1. Given an input of shouts, calculate the matching (the set of matched pairs) with Algorithm 3.1, and mark all the matched shouts as matched and all the other shouts as unmatched (lines 4-6).
2. Recursively check how many matches MM can achieve if the input shouts were matched asks and unmatched bids (line 8).
3. Recursively check how many matches MM can achieve if the input shouts were unmatched asks and matched bids (line 9).
4. Choose the minimum of the numbers from the last two steps as the extra number of matches MM can achieve (line 10).

³ Sales of identical goods or services are transacted at different prices is named *price discrimination* [7].

5. Cross match extra matchable shouts with the matched shouts in step 1: the ask in the first matched pair is rematched with the last extra matchable bid, while the bid in the pair is rematched with the last matchable ask, then the second matched pair with the second last extra matchable ask and bid, and so on until all extra matchable shouts are matched (lines 11-19).

Algorithm 3.2. MaximalMatching

Input: *Asks*: sorted in ascending order, *Bids*: sorted in descending order

Output: *Matching*

```

1 begin
2   Matching ← ∅;
3   if Asks = ∅ or Bids = ∅ then Return;
4   Matching ← EquilibriumMatching(Asks, Bids);
5   MatchedAsks ← all asks from Matching in ascending order;
6   MatchedBids ← all bids from Matching in descending order;
7   if (Bids \ MatchedBids) = ∅ or (Asks \ MatchedAsks) = ∅ then Return;
8   MM1 ← MaximalMatching(MatchedAsks, (Bids \ MatchedBids));
9   MM2 ← MaximalMatching((Asks \ MatchedAsks), MatchedBids);
10  ExtraNumberOfMatches ← Min(|MM1|, |MM2|);
11  I ← ExtraNumberOfMatches; N ← |Matching| + 1;
12  while I > 0 do
13    (Ask1, Bid1) ← read I-th in match from Matching;
14    Ask2 ← read N-th ask from Asks;
15    Bid2 ← read N-th bid from Bids;
16    Matching ← Matching ∪ {(Ask1, Bid2), (Ask2, Bid1)};
17    Matching ← Matching \ {(Ask1, Bid1)};
18    N ← N + 1; I ← I - 1;
19  end
20 end

```

Figure 1 shows a matching example of both EM and MM with the same set of shouts, where the numbers are the prices of shouts (other information is omitted), M indicates the last matchable pair with EM, and the arrowed lines link each matched pair. We can see that MM achieves two extra matches than EM does.

Complexity Analysis. MM is equivalent to finding a *maximum bipartite matching* in a bipartite graph $G = (V = (X^{ask}, X^{bid}), E)$, where E only contains one edge for each pair of ask a and bid b if $p(a) \leq p(b)$. Let $n_a = |X^{ask}|$, $n_b = |X^{bid}|$, and n_{em} and n_{mm} are the numbers of matches got with EM and MM, respectively. MM runs in $O(n_a \log n_a) + O(n_b \log n_b) + O((n_{em})^2)$ time in the worst case, where $O(n_a \log n_a)$ and $O(n_b \log n_b)$ are the complexities of sorting asks and bids (e.g. merge sort), and $O((n_{em})^2)$ is that of the rest of MM. The worst case condition for MM is that $n_{em} = \min(n_a, n_b) - 1$ holds for all EMs of MM, unless $\min(n_a, n_b) \leq 1$. So we can rewrite the complexity of MM as $O(\max(n_a, n_b) \log \max(n_a, n_b) + \min(n_a, n_b)^2)$. As reference, the best known

worst-case performance bipartite matching algorithm is *the Hopcroft-Karp algorithm*, which runs in $O(|E|\sqrt{n_a + n_b})$, where $|E| \geq (n_{em})^2$ in our model, time in the worst case [8].

4 Properties of Maximal Matching

In this section, we analyse a number of key properties of MM, especially market liquidity. In the following, given a set of shouts X , we use $M_{EM}(X)$ and $M_{MM}(X)$ to indicate the matching got from EM and MM, respectively, and use $N_{EM}(X)$ and $N_{MM}(X)$ for the corresponding number of matches.

4.1 Maximizing the Number of Transactions

We prove that MM indeed maximizes the number of transactions. Before the proof, we first give two lemmas about the connection between MM and EM.

Lemma 1. *Given a set of shouts X , let X_{EM} and X_{MM} are all the shouts included in $M_{EM}(X)$ and $M_{MM}(X)$, respectively, then $X_{EM} \subseteq X_{MM}$.*

Proof. Since MM first finds all the matches that can be found with EM, then checks if unmatched shouts could be matched with the matched ones, and if so, adds extra matchable shouts in the matching by changing which ask and bid to match, but without removing any already matched shouts, so all shouts in $M_{EM}(X)$ are included in $M_{MM}(X)$. \square

Lemma 2. *Given a set of shouts X , $N_{EM}(X) \leq N_{MM}(X) \leq 2 * N_{EM}(X)$.*

Proof. From Lemma 1, we can get $N_{MM}(X) \geq N_{EM}(X)$ because MM will return at least the number of matches returned from EM. Since in MM all the extra matchable shouts have to be matched with matched shouts in EM, so at most $N_{EM}(X)$ pairs of extra matchable shouts can be added. \square

Theorem 1. *Given a set of shouts X , $N_{MM}(X)$ is maximal.*

Proof. Assume that $N_{MM}(X)$ is not maximal, i.e. there is at least one more pair of $a_{i > N_{MM}(X)}$ and $b_{j > N_{MM}(X)}$ (i and j can be different) from the sorted asks and bids that should be included in the result matching (assume that the index starts from 1). We also know $p(a_i) > p(b_j)$ because of the sort and the jump condition in Algorithm 3.1 (line 6). Thus a_i and b_j have to be matched with another bid and ask with index $\leq N_{EM}(X)$, respectively. Now we check two conditions of $N_{MM}(X)$: one is $N_{EM}(X) \leq N_{MM}(X) < 2 * N_{EM}(X)$ and the other is $N_{MM}(X) = 2 * N_{EM}(X)$.

1. If $N_{EM}(X) \leq N_{MM}(X) < 2 * N_{EM}(X)$, the recursive call of MM will end up with either $p(a_k) > p(b_{N_{MM}(X)-k+2})$ or $p(a_{N_{MM}(X)-k+2}) > p(b_k)$ for some integer $k \in [1, e + 1]$, where $e = N_{MM}(X) - N_{EM}(X)$. If $p(a_k) > p(b_{N_{MM}(X)-k+2})$ for some k , based on the sort order, we also have $p(b_{N_{MM}(X)-k+2}) \geq p(b_j)$, so we get $p(a_k) > p(b_j)$, that is, b_j has to be

matched with some ask $a_{j' < k}$. However, each $a_{j' < k}$ has already been matched with another bid $b_{N_{MM}(X)-j'+1}$, and, based on the sort order, we know $p(b_{N_{MM}(X)-j'+1}) \leq p(b_{N_{MM}(X)-k+2})$, thus $p(a_k) > p(b_{N_{MM}(X)-j'+1})$, i.e. $b_{N_{MM}(X)-j'+1}$ cannot be matched with any ask ordered after a_{k-1} . If we want to match b_j , we have to remove some already matched bid. Similarly for $p(a_{N_{MM}(X)-k+2}) > p(b_k)$. Thus either a_i or b_j will not be matchable if $k = 1$, or some already matched shout(s) will be removed if $k > 1$, which contradicts the assumption.

2. If $N_{MM}(X) = 2 * N_{EM}(X)$, every shout with index $\leq N_{EM}(X)$ will be matched with another shout with index $> N_{EM}(X)$ in $M_{MM}(X)$. Thus we cannot match more shouts without losing matched ones in $M_{MM}(X)$, which again contradicts the assumption. \square

4.2 Maximizing Sell/Buy-Volume

Apart from the number of transactions, sell-volume and buy-volume are also measures of the liquidity of a marketplace. We show that MM not only improves both of them compared with EM, but also maximizes buy-volume and minimizes sell-volume compared with any matching method that gives the same number of matches as MM.

Definition 2. Given a matching M , the **sell-volume** is $\sum_{(a,b) \in M} p(a)$. The **buy-volume** is $\sum_{(a,b) \in M} p(b)$.

Given a set of shouts X , we have $\sum_{(a,b) \in M_{MM}(X)} p(a) \geq \sum_{(a,b) \in M_{EM}(X)} p(a)$ and $\sum_{(a,b) \in M_{MM}(X)} p(b) \geq \sum_{(a,b) \in M_{EM}(X)} p(b)$ because of Lemma 1. So MM will improve, if possible, sell-volume and buy-volume compared with EM.

Theorem 2. Given a set of shouts X , $\forall M \in \{M : |M(X)| = N_{MM}(X)\}$
 $\sum_{(a,b) \in M_{MM}(X)} p(a) \leq \sum_{(a,b) \in M(X)} p(a) \wedge \sum_{(a,b) \in M_{MM}(X)} p(b) \geq \sum_{(a,b) \in M(X)} p(b)$.

Proof. Given a set of shouts X , from the MM algorithm, we know that the first $N_{MM}(X)$ asks in ascending order and the first $N_{MM}(X)$ bids in descending order are matched. Thus the corresponding sell-volume and buy-volume will be minimal and maximal, respectively, given that the number of matches is $N_{MM}(X)$. \square

4.3 Maximizing Profit

In this section, we give the definition of the auctioneer's profit and analyse the impact of MM on that profit. Especially, we prove that MM maximizes the auctioneer's profit compared with any other matching method that matches the same number of pairs as MM.

Definition 3. Given matching M , the auctioneer's profit is the difference between the sum of the price of all bids and that of all asks in M :

$$U(M) = \sum_{(a,b) \in M} p(b) - \sum_{(a,b) \in M} p(a)$$

Theorem 3. *Given a set of shouts X , $U(M_{MM}(X)) \leq U(M_{EM}(X))$. Furthermore, $\forall M \in \{M : |M(X)| = N_{MM}(X)\} U(M_{MM}(X)) \geq U(M(X))$.*

Proof. Given a set of shouts X , let $m = N_{EM}(X)$ and $n = N_{MM}(X)$. From Lemma 2 we know $m \leq n$. From Lemma 1, if $m = n$, we have $U(M_{MM}(X)) = U(M_{EM}(X))$. If $m < n$, then

$$\begin{aligned} U(M_{EM}(X)) &= \sum_{i=1}^m (p(b_i) - p(a_i)) \\ U(M_{MM}(X)) &= \sum_{i=1}^m (p(b_i) - p(a_i)) + \sum_{i=m+1}^n (p(b_i) - p(a_i)) \end{aligned}$$

where b_i means the i th bid from X^{bid} in descending order and a_i is the i th ask from X^{ask} in ascending order w.r.t. their price. Since $p(b_i) \geq p(a_i)$ when $i \leq m$, and $p(b_i) < p(a_i)$ when $i > m$, we get $\sum_{i=1}^m (p(b_i) - p(a_i)) \geq 0$ and $\sum_{i=m+1}^n (p(b_i) - p(a_i)) < 0$. Thus $U(M_{MM}(X)) \leq U(M_{EM}(X))$.

Given the number of matches $N_{MM}(X)$, we can easily get $U(M_{MM}(X))$ is maximal from Theorem 2. \square

Although the auctioneer's profit with MM might be less than that with EM in the short term, the lost profit is actually used in liquidity maximization. High liquidity attracts traders, which will recursively increase liquidity and also lead to increased profit in the long term. We will show this from the experiments in Sect. 5.

4.4 Maximizing Social Welfare

Maximizing social welfare means maximizing the sum of traders' valuations of the item they are holding, i.e. matched buyers and unmatched sellers. Given bids B' that are matched and asks A' that are unmatched, then the term that we wish to maximize is: $\sum_{b \in B'} p(b) + \sum_{a \in A'} \delta p(a)$, where $\delta \in [0, 1]$ is a discount factor of the valuation of the goods that are not matched/traded. This is reasonable because, for instance, some commodities have an expiry date, e.g. ice cream and flowers, and there might be an inventory cost for un-traded goods.

Let B_{MM} and B_{EM} are the matched bids in MM and EM, respectively, we know from Lemma 1 that $B_{MM} \supseteq B_{EM}$, so we have $\sum_{b \in B_{MM}} p(b) \geq \sum_{b \in B_{EM}} p(b)$. If $\delta = 0$, MM maximizes social welfare. The proof is similar to the proof of Theorem 2.

5 Experimental Results

We have integrated the EM and MM specified in Sect. 3 in *jackaroo*⁴ which is a successful market under the CAT Tournament (Trading Agent Competition Market Design) platform. A CAT game is a simulation of exchange markets, and consists of buyers, sellers, and specialists. Each specialist operates and sets the rules for a single exchange market, and traders buy and sell goods in one of the available markets. In the CAT tournaments, the buyers and sellers are provided by the organizers, whereas specialists are designed by the entrants [9].

⁴ Achieved 3rd, 1st, and 2nd in CAT Tournament 2008, 2009, and 2010, respectively.

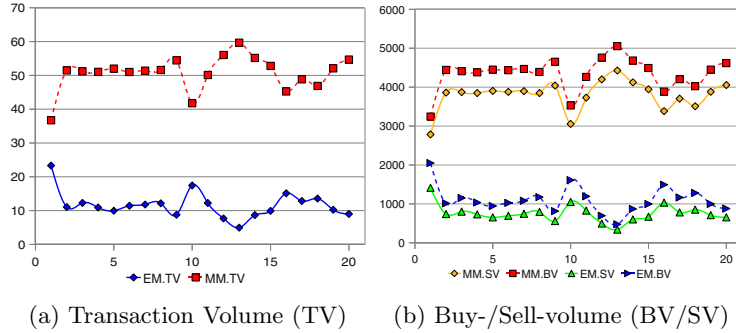


Fig. 2. Market Liquidity

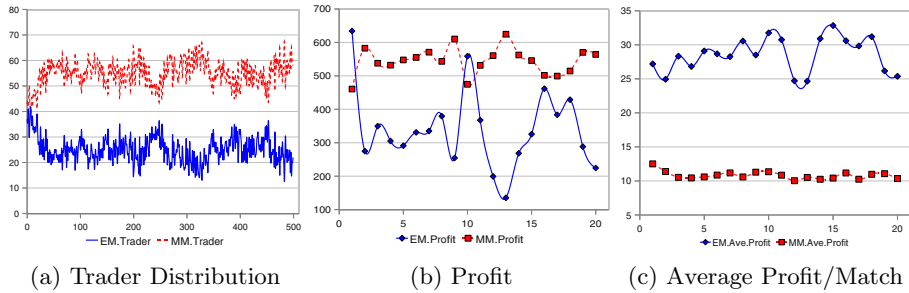


Fig. 3. Trader Distribution and Auctioneer's Profit

In our experiments, we use two markets (two variations of *jackaroo*) that use the same rules except that one uses EM and the other uses MM, and 80 profit-seeking traders with different intelligences to simulate a real market situation. We compare these two markets in 500 virtual days. In each day, each trader chooses only one market to exchange goods, and depending on the profit a trader got in that market, the trader might move to the other market on the next day.

Figures 2 and 3 show the main results related to our analysed properties in Sect. 4. Please note that Fig. 3 (a) shows information for each day, the rest use the average value of every 25 days. From Fig. 2 we can see that the transaction volume with MM is about 5 times that with EM, and both the buy-volume and sell-volume of the market with MM are about 4 times those of the EM market. As we mentioned in Sect. 4, good market liquidity will attract traders to the marketplace. This can be observed from the experiments. Figure 3 (a) shows the trader distribution in each day: in the very beginning, traders are equally distributed, but after 50 days, the MM market attracted most traders and it could maintain this until day 500. Because of trader attraction, the auctioneer's profit in the MM market is actually higher than that gained from the EM market (see Fig. 3 (b)), though the average profit for each transaction in the MM market is much smaller than that in the EM market (see Fig. 3 (c)). Note that, because the CAT Tournament platform was designed specifically for Trading

Agent Competition, it may not be the perfect testing environment for matching policies. Nonetheless, our experiments have provided a positive support to our theoretical results.

6 Conclusion

We have developed a new matching approach, called maximal matching, for a double auction market. We compared maximal matching with the most studied matching algorithm, equilibrium matching, and analysed its properties. The most distinctive property of maximal matching is liquidity maximization, which is a key factor in measuring the success of a marketplace. We also showed the advantages of maximal matching from experiments by using the CAT Tournament platform.

It is easy to see that a double auction mechanism with maximal matching is not necessarily incentive compatible. The essential feature of incentive compatibility is that the clearing price of a match does not depend on the bidding prices of the match. Maximal matching cannot guarantee this because the intersection of the clearing price ranges of all matches might be empty, while equilibrium matching can have incentive compatibility depending on how the uniform clearing price is chosen [4,5]. However, incentive compatibility is not compatible with most other desirable properties and is also very hard to achieve, especially in *dynamic/online double auction* (e.g. stock exchanges), where shouts are coming and leaving over time and there is more than one matching to search sequentially [3,10].

References

1. Friedman, D., Rust, J.: The Double Auction Market: Institutions, Theories, And Evidence. Westview Press, Boulder (March 1993)
2. Shubik, M.: A double auction market: Teaching, experiment, and theory. *Simul. Gaming* 36(2), 166–182 (2005)
3. Blum, A., Sandholm, T., Zinkevich, M.: Online algorithms for market clearing. *J. ACM* 53(5), 845–879 (2006)
4. McAfee, P.R.: A dominant strategy double auction. *Journal of Economic Theory* 56(2), 434–450 (1992)
5. Wurman, P.R., Walsh, W.E., Wellman, M.P.: Flexible double auctions for electronic commerce: theory and implementation. *Decis. Support Syst.* 24(1), 17–27 (1998)
6. Dash, R.K., Jennings, N.R., Parkes, D.C.: Computational-Mechanism design: A call to arms. *IEEE Intelligent Systems* 18(6), 40–47 (2003)
7. Nagle, T., Holden, R.: *The Strategy and Tactics of Pricing: a guide to profitable decision making*. Prentice Hall, Englewood Cliffs (November 2001)
8. Hopcroft, J.E., Karp, R.M.: A $n^{5/2}$ algorithm for maximum matchings in bipartite. In: *SWAT 1971: Proceedings of the 12th Annual Symposium on Switching and Automata Theory*, Washington, pp. 122–125. IEEE Computer Society, Los Alamitos (1971)
9. Cai, K., Gerding, E., Mcburney, P., Niu, J., Parsons, S., Phelps, S.: Overview of cat: A market design competition version 2.0 (2009)
10. Bredin, J., Parkes, D.C., Duong, Q.: Chain: a dynamic double auction framework for matching patient agents. *J. Artif. Int. Res.* 30(1), 133–179 (2007)