Mechanism Design for Double Auctions with Temporal Constraints *

Dengji Zhao[1,2] and **Dongmo Zhang**
[1] Intelligent Systems Lab
University of Western Sydney, Australia
{dzhao,dongmo}@scm.uws.edu.au

Laurent Perrussel
[2] IRIT
University of Toulouse, France
laurent.perrussel@univ-tlse1.fr

Abstract

This paper examines an extended double auction model where market clearing is restricted by temporal constraints. It is found that the allocation problem in this model can be effectively transformed into a weighted bipartite matching in graph theory. By using the augmentation technique, we propose a Vickrey-Clarke-Groves (VCG) mechanism in this model and demonstrate the advantages of the payment compared with the classical VCG payment (the Clarke pivot payment). We also show that the algorithms for both allocation and payment calculation run in polynomial time. It is expected that the method and results provided in this paper can be applied to the design and analysis of dynamic double auctions and futures markets.

1 Introduction

A double auction market allows multiple buyers and sellers to trade commodities simultaneously. Most modern exchange markets, e.g. the New York Stock Exchange, use double auction mechanisms. In a typical double auction market, buyers submit *bids* (buy orders) to the auctioneer (the market maker) offering the highest prices they are willing to pay for a certain commodity, and sellers submit *asks* (sell orders) to set the lowest prices they can accept for selling the commodity. The auctioneer collects the orders and tries to match them using certain market clearing policies in order to make transactions.

Although price is the major concern of market clearing in most double auction markets, other factors, such as quantity, quality and temporal constraints, are equally important in some market situations. For instance, a futures contract normally specifics not only the price of the underlying commodity but also quantity, quality and settlement date. Nevertheless, most real-world exchange markets are purely price-driven and most studies on double auctions are limited to a single-valued domain [Wilson, 1985; McAfee, 1992]. One reason is that some factors, e.g. quantity and quality, can be eliminated by standardising exchange commodities. However, those attributes with a continuous range or large number of varieties, are hard to standardise.

This paper considers an extension of the single-valued double auction model that allows traders to specify temporal constraints in their orders. Roughly speaking, an order is written in the form (p, t', t''), where p stands for the order price and $[t', t'']$ represents the time period when the commodity can be exchanged (not for the order itself). In this extension, a bid and an ask is matchable if and only if the bid price is no lower than the ask price and the intersection of their time constraints is non-empty. We found that the market clearing problem under this extension can be transformed into a weighted bipartite matching. This allows us to use some standard techniques in graph theory, such as augmentation, for the design and analysis of the mechanisms. We prove that an allocation for the double auction is efficient if and only if it corresponds to a maximum weighted bipartite matching of the graph encoding the incoming orders. Based on that, we develop an efficient and dominant-strategy incentive-compatible double auction mechanism, i.e. a VCG mechanism [Groves, 1973]. Remarkably, our payment can be implemented much faster than the classical VCG payment, known as *Clarke pivot payment*, while resulting in the same payments, because it directly uses the abridging and replacing paths generated during the allocation process rather than recall the allocation algorithm as Clarke pivot payment does.

It is worth mentioning that during the last decade many researchers started to look at the mechanism design problem for dynamic environments where traders are arriving and departing dynamically, referred to as *online mechanism design* [Parkes, 2007]. To model the dynamics, temporal information is also used. Although the meaning of the temporal information of a trader's type in the online setting is different from that in our setting, a trader's type is modelled in the same way in both settings [Blum *et al.*, 2006; Bredin *et al.*, 2007]. Therefore, the mechanism in our model also provides an optimal (offline) solution for a corresponding dynamic market. Such an optimal solution can be directly used for calculating the competitive ratio of an online market-clearing algorithm. Moreover, although orders arrive dynamically, the alternating paths are relatively stable and therefore can be used, for example, to identify potential good orders to find more efficient allocations in an online setting.

This paper is organised as follows. In Section 2 we briefly introduce our market model and related concepts. In Section 3, we introduce a graphic representation for market situa-

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

tions and transfer the market clearing problem into a weighted bipartite matching. In Section 4, we concentrate on the design of an allocation algorithm and a payment algorithm, and prove their desirable properties. We conclude in Section 5 with a brief discussion for the future work.

2 The Model

Consider a double auction market, in which a set B of **buyers** and a set S of **sellers** trade one commodity simultaneously. Buyers and sellers are **traders**. Let $T = B \cup S$ and assume that the traders are independent and $B \cap S = \varnothing$. We also assume that each seller and each buyer supplies and demands a single unit of the commodity.

Each trader $i \in T$ has a privately observed **type** $\theta_i = (v_i, s_i, e_i)$, where v_i, s_i and e_i are non-negative real numbers, v_i is the trader's valuation of a single unit of the commodity, and s_i and e_i are the starting point and the ending point of the time constraint $[s_i, e_i]$. If trader i is a buyer, i obtains **utility** $v_i - p$ if i receives a unit of the commodity within the time interval $[s_i, e_i]$ and pays p; i obtains zero utility if i pays nothing and does not receive the commodity within the time period. Similarly, if i is a seller, i obtains utility $p - v_i$ if i successfully sells a unit of the commodity within the time period $[s_i, e_i]$ and receives payment p; i obtains zero utility if i fails to sell the commodity within the time period and no payment is made.

Let $\theta = (\theta_i)_{i \in T}$ denote the type profile where θ_i is the type of trader i. θ_{-i} means the type profile of all traders except trader i. Note that we treat a type profile as a vector of types rather than a set of types. Let Θ_i be the set of all possible types of trader i, and we write $\Theta = (\Theta_i)_{i \in T}$.

Since we will focus on direct-revelation mechanisms, we assume that traders directly report their types to the auctioneer as their **orders** [Myerson, 2008]. Traders do not necessarily truthfully report their types but no early-start and no late-end misreports are permitted. Formally, let $\theta_i = (v_i, s_i, e_i)$ be trader i's type and $\hat{\theta}_i = (\hat{v}_i, \hat{s}_i, \hat{e}_i)$ be the trader's **report**. We assume that $[\hat{s}_i, \hat{e}_i] \subseteq [s_i, e_i]$. The intuition behind the assumption is that no trader would report a temporal constraint that might give him negative utility. Let $R(\theta_i)$ be the set of all permitted reports from trader i given his type θ_i, $R(\Theta_i) = \bigcup_{\theta_i \in \Theta_i} R(\theta_i)$ be the set of all possible reports from i, and $R(\Theta) = (R(\Theta_i))_{i \in T}$.

Given traders' reports $\theta \in R(\Theta)$, an **ask** $\theta_i = (v_i, s_i, e_i)$ (means $i \in S$) and a **bid** $\theta_j = (v_j, s_j, e_j)$ (means $i \in B$) are **matchable** if and only if $v_i \le v_j$ and $[s_i, e_i] \cap [s_j, e_j] \neq \varnothing$. That is, the bid's valuation is no less than the ask's valuation, and the intersection of their time constraints is not empty.

An **allocation policy** $\pi = (\pi_i)_{i \in T}$ is a function that assigns 0 or 1 to each trader, given traders' reports $\hat{\theta} \in R(\Theta)$. For a trader i, if $\pi_i(\hat{\theta}) = 1$ we say i wins; otherwise i loses. An allocation determines whose order is granted for a transaction.

A **payment policy** $x = (x_i)_{i \in T}$ is a function that assigns a real number to each trader given an input of traders' reports $\hat{\theta} \in R(\Theta)$, i.e. $x_i(\hat{\theta}) \in \mathbb{R}$ for all $i \in T$.

Definition 1. *A* **double auction mechanism** *on Θ is a pair (π, x), where π is an allocation policy and x is a payment policy.*

Following the standard definition, we say that an auction mechanism (π, x) is **efficient** if π maximizes

$$\sum_{i \in B \& \pi_i(\theta)=1} v_i + \sum_{i \in S \& \pi_i(\theta)=0} v_i.$$

for any type profile $\theta = ((v_i, s_i, e_i))_{i \in T}$.

We say that an auction mechanism is **dominant-strategy incentive-compatible**, i.e. truthful, if for each trader, reporting his true type is his dominant strategy.

There are a number of alternatives to characterise truthfulness in an auction mechanism. We will use one of them in this paper based on [Parkes, 2007; Nisan, 2007]. To describe it, we need the following two auxiliary concepts [Parkes, 2007].

For each trader i, we define a partial order $\preceq_i$ on $R(\Theta_i)$:

$$\hat{\theta}'_i \preceq_i \hat{\theta}''_i \text{ iff } \begin{cases} v'_i \ge v''_i \;\&\; [s'_i, e'_i] \subseteq [s''_i, e''_i], & \text{if } i \in S \\ v'_i \le v''_i \;\&\; [s'_i, e'_i] \subseteq [s''_i, e''_i], & \text{if } i \in B \end{cases}$$

where $\hat{\theta}'_i = (v'_i, s'_i, e'_i)$ and $\hat{\theta}''_i = (v''_i, s''_i, e''_i) \in R(\Theta_i)$.

We say that an allocation policy π is **monotonic** if, for each $i \in T$, $\pi_i(\hat{\theta}_i, \hat{\theta}_{-i}) = 1$ implies $\pi_i(\hat{\theta}'_i, \hat{\theta}_{-i}) = 1$ whenever $\hat{\theta}_i \preceq_i \hat{\theta}'_i$.

Definition 2. *Given a monotonic policy π and traders' reports $\hat{\theta} \in R(\Theta)$, the **critical value** of trader i of type $\theta_i = (v_i, s_i, e_i)$ is defined as*

$$c(\theta_i, \hat{\theta}_{-i}) = \begin{cases} \sup\{v'_i : & (v'_i, s_i, e_i) \in R(\theta_i) \wedge \\ & \pi_i((v'_i, s_i, e_i), \hat{\theta}_{-i}) = 1\}, \quad \textit{if } i \in S \\ \inf\{v'_i : & (v'_i, s_i, e_i) \in R(\theta_i) \wedge \\ & \pi_i((v'_i, s_i, e_i), \hat{\theta}_{-i}) = 1\}, \quad \textit{if } i \in B \end{cases}$$

It is undefined if no such v'_i exists.

Now we are ready to describe a characterisation of truthfulness, which will be used in Section 4. Theorem 1 is based on Theorem 9.36 in [Nisan, 2007] for a single-valued domain and on [Parkes, 2007] for a single-valued online domain. The proof is omitted here as it is similar to the above mentioned theorems.

Theorem 1. *A double auction mechanism (π, x) is dominant-strategy incentive-compatible if and only if:*

- *π is monotonic.*
- *every winning seller (buyer) is paid (pays) his critical value, and the payment is 0 for losing traders.*

3 Graph Representation

As assumed in the previous section, each trader has only one unit of a commodity to sell or buy. Transaction must be made in pairs: a seller can only sell his good to a unique buyer, assuming their orders are matchable. This means that to allocate the goods in a double auction is to find matchings between buy orders and sell orders. In such a case we can transform the allocation problem into a matching problem in graph theory. As a result an *efficient* allocation corresponds to a maximum-weighted bipartite matching. We will first review some concepts related to bipartite matching [West, 2000], encode incoming orders in a bipartite graph, and then show some special properties related to the encoding.

Definition 3. *A graph $G = (V, E)$ is a* **bipartite graph** *if the vertex set V consists of two disjoint subsets X and Y, and no edge has both end points in the same subset. For explicitness, we denote the graph as $G = ((X, Y), E)$.*

Definition 4. *Given a traders' report $\theta \in R(\Theta)$, we call $\mathbf{G}_\theta = ((\mathbf{S}^\theta, \mathbf{B}^\theta), \mathbf{E})$ a* **bipartite graph generated from** *θ if*

- *$S^\theta = \{\theta_i : i \in S\}$ and $B^\theta = \{\theta_i : i \in B\}$,*
- *$E = \{(\theta_i, \theta_j) : \theta_i$ and θ_j is matchable$\}$.*

Definition 5. *Given a graph G, a **matching** M in G is a set of pair-wise non-adjacent edges, i.e. no two edges share a common vertex. The **size** of M is denoted by $|M|$. A vertex is **matched** if it is incident to an edge in the matching. Otherwise the vertex is **free**.*

Given a matching M,

- an M**-alternating path** is a path in which the edges belong alternatively to M and not to M.
- an M**-augmenting path** is an M-alternating path whose endpoints are free.
- an M**-abridging path** is an M-alternating path whose first edge and last edge are in M.
- an M**-replacement path** is an M-alternating path where one of the endpoints is free and one of the ending edges is in M.

A path is **simple** if it has no repeated vertices. In the rest of this paper, we will only consider simple paths.

Figure 1 shows an example of bipartite representation of eight different type reports. Lines and dashed lines indicate matched edges and free edges respectively, and dots and circles denote matched vertices and free vertices respectively. The value beside each vertex is its valuation. Temporal information is not shown in the graph. It is clear that path `(3,10,2,9)` is an augmenting path, path `(2,10,4,7)` is an abridging path, and path `(2,10,4,7,5)` is a replacement path.

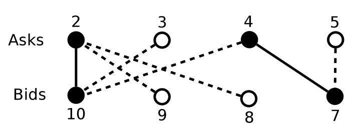

Figure 1: Example of Alternating Paths

Given a matching M, we can use an M-augmenting path p to augment M by changing all matched edges in p to be free and all the free edges to be matched. By contrast an M-abridging path can be used in the same way to abridge M. Consequently, $|M|$ will increase (decrease) by one with one augmenting (abridging) process. An M-replacement path can be used to replace a bid or an ask in M without changing the status of the other vertices.

Definition 6. *An allocation policy π is* **feasible** *if for any traders' reports $\theta \in R(\Theta)$, there is a matching M in the bipartite graph generated from θ such that M exactly covers $\{\theta_i : \pi_i(\theta) = 1\}$.*

It follows that any matching in a bipartite graph generated from traders' reports uniquely determines a feasible allocation. In the rest of this paper, we will only consider feasible allocation policies.

Definition 7. *Given bipartite graph G_θ, an edge e between $\theta_i = (v_i, s_i, e_i)$ and $\theta_j = (v_j, s_j, e_j)$, where $i \in S$ and $j \in B$, we define the **weight** of e as $w(e) = v_j - v_i$. For any set of edges $E' \subseteq E$, the weight of E' is defined as*

$$w(E') = \sum_{e \in E'} w(e).$$

*The **weight increase** of an M-alternating path p is the total weight of free edges in p minus that of matched edges in p:*

$$\Delta(p) = w(P - M) - w(P \cap M),$$

where P is the set of all edges in p.

If p is an M-augmenting, M-abridging, or M-replacement path, then $\Delta(p)$ is the net change in the weight of the matching after augmenting, abridging, or replacing by p: $w(M \oplus p) = w(M) + \Delta(p)$, where $M \oplus p \equiv M \oplus P$, P is the set of all edges in p, and $\oplus$ is the **symmetric difference** operator on sets: $A \oplus B = (A \cup B) \setminus (A \cap B)$.

Lemma 1. *Given G_θ, a matching M in G_θ, and an M-alternating path p, $\Delta(p)$ is equal to*

- *the valuation of the bid minus that of the ask of the endpoints of p, if p is an augmenting path.*
- *the valuation of the ask minus that of the bid of the endpoints of p, if p is an abridging path.*
- *the valuation of the free (matched) endpoint minus that of the matched (free) endpoint of p when the endpoints are bids (asks), if p is a replacement path.*

We will not provide the proof of Lemma 1 which follows the weight definition of the edges.

4 Efficient and Truthful Policy Design

In order to design a double auction that is both efficient and truthful, by Theorem 1, we need to find an efficient and monotonic allocation policy, and a payment policy that calculates the critical value of each winning trader. Inspired by the similarity between this allocation problem and the weighted matching in a bipartite graph, we first transform the model into a bipartite graph. Within this graph, we show how to efficiently use the well established methods from bipartite matching in the allocation policy, and how to calculate critical payment without running the allocation policy again.

4.1 Efficient & Monotonic Allocation Policy

With the above graph encoding of traders' reports, we designed an *efficient* allocation policy by adopting the *maximum-weighted bipartite matching* that constructs a maximum-weighted matching by beginning with the empty matching and repeatedly performing augmentations using augmenting paths of maximum weight increase until a maximum-weighted matching is achieved [Tarjan, 1983; Kozen, 1991]. The resulting allocation policy is called **Maximum-weighted Bipartite Matching Allocation** (*MBM Allocation*), which seeks an allocation that maximises social welfare of any reports θ, by first representing θ in a bipartite graph G_θ, and then applying modified *maximum-weighted bipartite matching* to get a maximum-weighted matching M which determines all winning reports.

We added a more detailed path selection rule in the *maximum-weighted bipartite matching* in order to achieve the *monotonicity* property. The rule is based on the order $\preccurlyeq_p$ defined for augmenting paths. Let a sequence of vertices $\theta_1 \circ ... \circ \theta_n$ denote an augmenting path of length n, which starts from ask θ_1 and ends in bid θ_n. We define $\preccurlyeq_p$ on all augmenting paths based on their endpoints:

$$\theta_1 \circ ... \circ \theta_n \preccurlyeq_p \theta'_1 \circ ... \circ \theta'_m \text{ iff}$$
$$(v'_1, v_n, s'_1, e_1, s'_m, e_n) \leqslant_s (v_1, v'_m, s_1, e'_1, s_n, e'_m),$$

where $\leqslant_s$ is the lexicographic order of two equal length sequences of real numbers: $(r^1_1, ..., r^1_n) \leqslant_s (r^2_1, ..., r^2_n)$ iff $\exists_{1\le j\le n}(r^1_j \le r^2_j \wedge \forall_{1\le k<j}(r^1_k = r^2_k))$. We will use $\preccurlyeq_p$ in *MBM Allocation* to distinguish augmenting paths that have the same weight increases.

Maximum-weighted Bipartite Matching Allocation:

Initialization:

- Encode reports θ in bipartite graph G_θ.
- Set the result matching $M = \emptyset$ for G_θ.

Recursion:

- $AugPath = \{p : \Delta(p) > 0$ and p is an M-augmenting path$\}$.
- $MaxAugPath = \arg\max_{p\in AugPath} \Delta(p)$.
- If $MaxAugPath = \emptyset$, stop recursion.
- Otherwise, let $\hat{p} \in MaxAugPath$ s.t. $p \preccurlyeq_p \hat{p}$ for any $p \in MaxAugPath$, and $M = M \oplus \hat{p}$.

Output:

- All reports covered by M win and all the rest lose.

Theorem 2. *Maximum-weighted Bipartite Matching Allocation is efficient.*

We prove Theorem 2 in the Appendix. Here we show one essential lemma used in the proof. In the rest of this paper, π denotes *MBM Allocation*.

Lemma 2. *Maximum-weighted Bipartite Matching Allocation is efficient if and only if the* maximum-weighted bipartite matching *maximizes the weight of the matching.*

Proof. The weight of the matching is $\sum_{\pi_i(\theta)=1\wedge i\in B} v_i - \sum_{\pi_i(\theta)=1\wedge i\in S} v_i$, which is equal to $\left(\sum_{\pi_i(\theta)=1\wedge i\in B} v_i + \sum_{\pi_i(\theta)=0\wedge i\in S} v_i\right) - \sum_{i\in S} v_i$. $\sum_{i\in S} v_i$ is fixed, so if the weight of the matching is maximised, then $\sum_{\pi_i(\theta)=1\wedge i\in B} v_i + \sum_{\pi_i(\theta)=0\wedge i\in S} v_i$ is also maximized, and vice versa. □

Theorem 3. *Maximum-weighted Bipartite Matching Allocation is monotonic.*

Although we added a specific path selection rule based on $\preccurlyeq_p$ to avoid randomisation of *MBM Allocation* in most cases, there is still one situation where $\preccurlyeq_p$ cannot help. When two types are the same and two augmenting paths of maximum positive weight increase start from them and end in the same vertex, then $\preccurlyeq_p$ cannot separate these two paths clearly, i.e. both of them have a chance of being selected but none of them are guaranteed. Thus we assume that all type reports of sellers (buyers) are different. Note that there might be more than one augmenting path with the same endpoints, but this does not affect the deterministic property of *MBM Allocation*. The proofs of all the theorems are given in the Appendix.

4.2 Truthful Payment Policy

We have found an efficient allocation policy, *MBM Allocation*, and proved its monotonicity property which is one of the two iff conditions to satisfy truthfulness. What is left is to calculate the critical value for each winning trader.

Obviously, it is not practical to calculate the critical value as it's defined in Definition 2. Here we propose another approach which is inspired by the reverse of *MBM Allocation*. A type θ_i is matched because there is an augmenting path of maximum positive weight increase ending with θ_i in some round of the matching procedure. Therefore, if a type does not satisfy the above condition, it will not be matched. This is the basis of our payment policy which is seeking the least violation condition for each winning type, i.e. the edge condition between winning and losing.

Given traders' reports θ, if $\pi_i(\theta) = 1$, the payment for trader i, $x_i(\theta)$, is defined in terms of abridging and replacement paths starting from θ_i in the following, which is called **Min-Max Payment** (*MM Payment*). $x_i(\theta) = 0$ if $\pi_i(\theta) = 0$.

Min-Max Payment:

$$x_i(\theta) = \begin{cases} \min_{p\in D\cup R} v(ending(p)), & \text{if } i \in S \\ \max_{p\in D\cup R} v(ending(p)), & \text{if } i \in B \end{cases}$$

where

- D is a set of all abridging paths starting from θ_i,
- R is a set of all replacement paths starting from θ_i,
- and $v(ending(p))$ is the valuation of the ending vertex, the endpoint other than θ_i, of path p.

For each winning ask, *MM Payment* gives the minimum valuation such that, if the ask's valuation were greater than or equal to that minimum, it can be removed from the matching to (weakly) increase the weight of the matching, while for each winning bid, the payment is the corresponding maximum. The set D gives all possible ways to remove θ_i by abridging, while the set R gives all possible ways to substitute a free vertex for θ_i. Note that set D does not necessarily contain the path that was used to match θ_i, as the path can be changed with other augmentations after adding θ_i.

Theorem 4. *Given bipartite graph G_θ and a winning type $\theta_i = (v_i, s_i, e_i)$ determined by MBM Allocation, Min-Max Payment $x_i(\theta)$ is equal to critical value $c(\theta_i, \theta_{-i})$.*

Another appealing property of Min-Max payment is its independence from the allocation algorithm. We show that Min-Max payment results in the same payments as the most desirable VCG payment (Clarke pivot payment), but it does not require the recall of the allocation algorithm. Clarke pivot payment is defined as $x_i(\theta) = V^{\pi}(\theta_{-i}) - V^{\pi}_{-i}(\theta)$, where $V^{\pi}(\theta)$ is the social value given traders' report profile θ and the allocation policy π, and $V^{\pi}_{-i}(\theta)$ is the social value without counting trader i.

Lemma 3. *Given traders' report θ and efficient and monotonic allocation policy π, for each trader i, Min-Max payment $x_i^{MM}(\theta)$ is equal to Clarke pivot payment $x_i^C(\theta)$.*

Proof sketch. We need to prove that for each winning type θ_i if we remove θ_i from the maximum-weighted matching M of bipartite graph G_θ by using the path p that gives $x_i^{MM}(\theta)$, the result matching M' is also maximum-weighted in $G_{\theta_{-i}}$. By contradiction, assume that M' is not maximum-weighted, we will conclude either M is not maximum-weighted or path p contradicts the definition of Min-Max payment. □

Corollary 1. *Double auction mechanism (MBM Allocation, MM Payment) is efficient, dominant-strategy incentive-compatible and individual-rational, i.e. traders never get negative utility.*

Figure 2 shows an example of the double auction we have defined, where the number beside each vertex is the valuation of the vertex and the value inside parentheses is the payment.

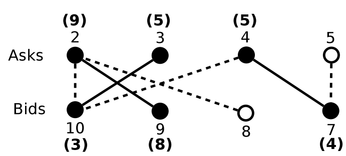

Figure 2: MBM Allocation and MM Payment

4.3 Computational Complexity

We further show that both our allocation policy and payment policy can be implemented in polynomial time and, more importantly, our payment can be implemented much faster than Clarke pivot payment.

Theorem 5. *Let n be the number of traders' reports. MBM Allocation can be implemented in time $O(n^3)$, and Min-Max Payment can be implemented in time $O(n^3)$.*

This result is significant because, to the best of our knowledge, the implementations of Clarke pivot payment cannot avoid the recall of the allocation algorithm [Nisan and Ronen, 1999; Sandholm, 2003]. In other words, for each winning report θ_i, π needs to search another allocation on the remaining reports θ_{-i}. Therefore, it will take $O(n)$ times of the allocation time in this model, i.e. $O(n^4)$ with *MBM Allocation*.

5 Conclusion

We have developed an efficient and truthful double auction mechanism (i.e. a VCG mechanism) in a model where each trader's type consists of a valuation of a commodity and a time period that constrains when the commodity can be exchanged. This mechanism is characterised by an allocation policy and a payment policy. By encoding the model in a bipartite graph, we efficiently adapted the *maximum-weighted bipartite matching* to get an efficient and monotonic allocation policy. We also developed a truthful payment policy that can be implemented faster than Clarke pivot payment while resulting in the same payments as Clarke pivot payment.

Myerson *et al.* [1983] proved that there is no efficient, incentive-compatible and individual-rational bilateral trade without outside subsidies, i.e. a market with our mechanism will run in deficit. To avoid this deficit, we need to compromise between efficiency and truthfulness. There are two possible remedies: either relaxing efficiency, or giving up incentive compatibility, as investigated by McAfee [1992] and Wurman *et al.* [1998] in single-valued domains. Finding how these compromises can lead to a realistic mechanism under our model is worth further investigation.

Appendix: Proofs of Theorems

Proof of Theorem 2: In order to prove Theorem 2, by Lemma 2, we shall prove that the *maximum-weighted bipartite matching* indeed gives a maximum-weighted matching. To do that, we need the two verified properties of the *maximum-weighted bipartite matching* given in Claim 1 and 2 [Tarjan, 1983]. This is one of the advantages we gained by encoding the model in a graph. We will skip the proofs of the following two claims.

Claim 1. *Given graph G, let M be a matching of size k of maximum weight among all matchings of size k in G. If we augment M by an augmenting path of maximal weight increase, then we obtain a matching of size $k+1$ of maximum weight among all matchings of size $k+1$ in G.*

Claim 2. *The maximum-weighted bipartite matching will augment along augmenting paths of successively nonincreasing* weight increase.

By Claim 1, the *maximum-weighted bipartite matching* will give a matching M_k of size k of maximum weight among all matchings of size k after k augmentations. By Claim 2, M_k is also maximum-weighted among all matchings of size at most k if the weight increase at the k-th augmentation is positive. Therefore, the matching it gives until there is no augmenting path of positive weight increase is maximum-weighted among all matchings. □

Proof of Theorem 3: By contradiction, without loss of generality, assume that $\pi_i(\theta) = 1$ and $\pi_i(\theta'_i, \theta_{-i}) = 0$ for some bids $\theta_i \preceq_i \theta'_i$. Let θ_i be matched in round k of $\pi(\theta)$, i.e some augmenting path ending with θ_i is of maximal weight increase in round k. Since θ and θ'_i are both not matched before round k, so the matchings are the same in both $\pi(\theta)$ and $\pi(\theta'_i, \theta_{-i})$ after any round $< k$. Let $\theta_m \circ ... \circ \theta_i$ be the augmenting path of maximal weight increase selected in round k of $\pi(\theta)$. Since $\theta_i \preceq_i \theta'_i$, $\theta_m \circ ... \circ \theta'_i$ is an augmenting path in round k of $\pi(\theta'_i, \theta_{-i})$ and $\theta_m \circ ... \circ \theta_i \preccurlyeq_p \theta_m \circ ... \circ \theta'_i$. Moreover, in round k, all augmenting paths in $\pi(\theta'_i, \theta_{-i})$, except those that end with θ'_i, are also augmenting paths in $\pi(\theta)$. Thus, in round k of $\pi(\theta'_i, \theta_{-i})$, for any augmenting path p that does not end with θ'_i, $p \preccurlyeq_p \theta_m \circ ... \circ \theta'_i$, and all the rest end with θ'_i. Therefore, an augmenting path ending with θ'_i should be selected in round k of $\pi(\theta'_i, \theta_{-i})$, which contradicts the assumption. □

Proof of Theorem 4: The proof needs the following two claims which can be found in [Kozen, 1991; Blum *et al.*, 2006].

Claim 3. *Given two matchings M and M', $M \oplus M'$ consists of a collection of vertex-disjoint alternating paths and even length cycles.*

Claim 4. *Given two matchings M and M', a vertex v is an endpoint of a path in $M \oplus M'$ if and only if it is matched in either M or M' but not both.*

Now we are ready to prove Theorem 4. Without loss of generality, assume $\theta_i = (v_i, s_i, e_i)$ is a winning ask, and let $x^i = x_i(\theta)$ and $c^i = c(\theta_i, \theta_{-i})$. To prove $x^i = c^i$, by the definition of c^i, we need to show that for any $\theta_i' = (v_i', s_i, e_i)$:

1. $\forall_{\theta_i':\pi_i(\theta_i',\theta_{-i})=1}(v_i' \leq x^i)$.
2. $\exists_{x^i-\delta<v_i'<x^i}(\pi_i(\theta_i', \theta_{-i}) = 1)$ for any $\delta > 0$.

Let M be the matching of $\pi(\theta)$ and M' be that of $\pi(\theta_i', \theta_{-i})$. We will prove these two conditions one by one blow.

Part I: By contradiction, assume that $\pi_i(\theta_i', \theta_{-i}) = 1$ and $v_i' > x^i$. Let A_M (B_M) be all the matched asks (bids) in M, and $A_{M'}$ ($B_{M'}$) be all the matched asks (bids) in M'. Since π is monotonic and M and M' are maximum-weighted, it follows that all matched asks in M except for θ_i must be matched in M', i.e. $A_M \setminus \{\theta_i\} \subset A_{M'}$, and all the matched bids in M' must be matched in M, i.e. $B_{M'} \subseteq B_M$. Thus inequalities $|A_M| - 1 < |A_{M'}|$ and $|B_{M'}| \leq |B_M|$ hold. Moreover, $|A_M| = |B_M|$ and $|A_{M'}| = |B_{M'}|$, so we get $|M| = |M'|$, $B_M = B_{M'}$, and $A_M \setminus \{\theta_i\} = A_{M'} \setminus \{\theta_i'\}$. Therefore, by Claim 3 and 4, there is only one alternating path $p_{only} = \theta_i \circ ... \circ \theta_i'$ in $M \oplus M'$, and all the rest are cycles. If all vertices reachable from θ_i through M-abridging or M-replacement paths are also reachable from θ_i' through M'-abridging or M'-replacement paths, then, since $v_i' > x^i$, there is at least one M'-abridging or M'-replacement path of positive weight increase by which we can remove θ_i' to increase the weight of the matching, which contradicts the choice of M'.

Let us prove that the above reachability condition holds. (1). For any vertex v except for θ_i (θ_i') in p_{only}, the path between θ_i (θ_i') and v is either an abridging or a replacement path with respect to M (M'). (2). Any vertex v' not in p_{only} that is reachable from θ_i by an abridging or replacement path p is also reachable from θ_i' through the same type of path p'. Since p must be connected with p_{only} and for any edge $e \in p$ and $e \notin p_{only}$, if $e \in M$ and $e \notin M'$, there must be an even length cycle that contains e in $M \oplus M'$, and vice versa, i.e. if e connects vertices v_1 and v_2 in p, there is always a corresponding edge or path connecting v_1 and v_2 in p'. For instance, Figure 3 shows one alternating path `(a,b,c,d,e)` and a cycle `(h,i,j,k)` of $M \oplus M'$: thin lines and thick lines belong to M and M' respectively, while the double line between f and g is in both matchings and dashed lines are free. It is easy to see that all vertices reachable from a through a M-augmenting or M-replacement path is also reachable from e by a corresponding path with respect to M'.

Part II: To prove the second condition, we will prove $\pi_i(\theta_i', \theta_{-i}) = 1$ for any $v_i' < x^i$. By contradiction, assume that $v_i' < x^i$ and $\pi_i(\theta_i', \theta_{-i}) = 0$. By Claim 4 there is a path $p_{\theta_i} \in M \oplus M'$ starting from θ_i and ending with θ_n in either M or M'. Since $x^i \leq v_n$ by the definition of x^i and $v_i' < x^i$ as we assumed, we can substitute θ_i' for θ_i in p_{θ_i} to get an M'-alternating path $p_{\theta_i'}$. If θ_n is matched in M, then $p_{\theta_i'}$ is an M'-augmenting path and by Lemma 1 $\Delta(p_{\theta_i'}) = v_n - v_i' > 0$, which contradicts the choice of M'. Thus θ_n is a matched ask in M', and $p_{\theta_i'}$ is an M'-replacement path. Since M' is a maximum-weighted, by Lemma 1 $\Delta(p_{\theta_i'}) = v_n - v_i' \leq 0$. Put all results together, we get contradiction $v_n \leq v_i' < x^i \leq v_n$. □

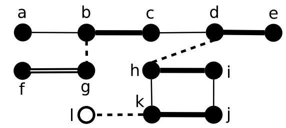

Figure 3: Reachability Example

Proof of Theorem 5: Bipartite graph representation of the reports takes at most $n^2/4$ time by checking each pair of ask and bid, so there will be at most $m = n^2/4$ edges. According to [Galil, 1986], finding an augmenting path of maximal weight increase can be solved by Dijkstra's algorithm taking $O(m+n \log n)$ time. There are at most $n/2$ rounds, so *MBM Allocation* can be implemented in time $O(n^3)$. For each winning type, *MM Payment* can be done by depth-first or breadth-first search which takes $O(n + m)$ time. There are at most n winning types, so *MM Payment* can also be implemented in time $O(n^3)$. □

References

[Blum *et al.*, 2006] A. Blum, T. Sandholm, and M. Zinkevich. Online algorithms for market clearing. *J. ACM*, 2006.

[Bredin *et al.*, 2007] J. Bredin, D. C. Parkes, and Q. Duong. Chain: a dynamic double auction framework for matching patient agents. *J. Artif. Int. Res.*, 30(1):133–179, 2007.

[Galil, 1986] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Comput. Surv.*, 18:23–38, March 1986.

[Groves, 1973] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–31, July 1973.

[Kozen, 1991] D. Kozen. *The Design and Analysis of Algorithms*. Springer-Verlag, 1991.

[McAfee, 1992] R. P. McAfee. A dominant strategy double auction. *Journal of Economic Theory*, 56(2):434–450, April 1992.

[Myerson and Satterthwaite, 1983] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265–281, April 1983.

[Myerson, 2008] R. B. Myerson. revelation principle. In Steven N. Durlauf and Lawrence E. Blume, editors, *The New Palgrave Dictionary of Economics*, 2008.

[Nisan and Ronen, 1999] N. Nisan and A. Ronen. Algorithmic mechanism design, *Games and Economic Behavior*, pages 129–140, 1999.

[Nisan, 2007] N. Nisan. Introduction to mechanism design (for computer scientist). In *Algorithmic Game Theory*. Cambridge University Press, 2007.

[Parkes, 2007] D. C. Parkes. Online mechanisms. In *Algorithmic Game Theory*. Cambridge University Press, 2007.

[Sandholm, 2003] T. Sandholm. Making markets and democracy work: A story of incentives and computing. In *AAAI*, pages 1649–1671, 2003.

[Tarjan, 1983] R. E. Tarjan. *Data structures and network algorithms*. Society for Industrial and Applied Mathematics, 1983.

[West, 2000] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2000.

[Wilson, 1985] R. Wilson. Incentive Efficiency of Double Auctions. *Econometrica*, 53(5):1101–1115, 1985.

[Wurman *et al.*, 1998] P. R. Wurman, W. E. Walsh, and M. P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation, *Decision Support Systems*, 1(24), 1998.