

# Mechanism Design for Double Auctions with Temporal Constraints

Dengji Zhao<sup>1,2</sup>

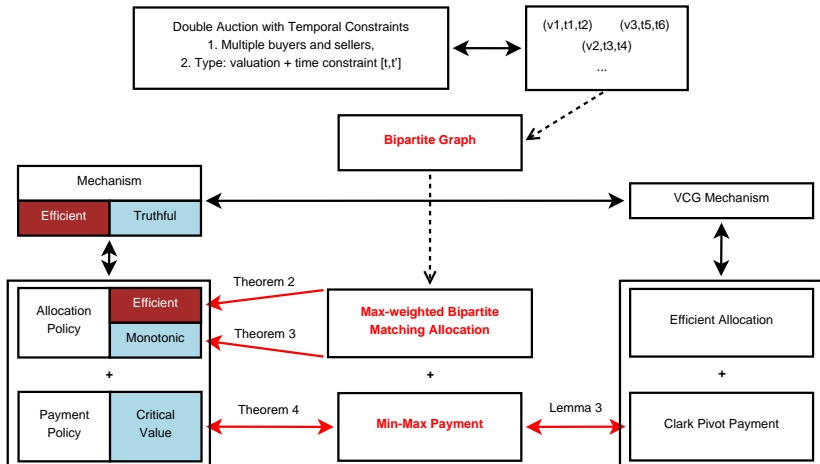
Dongmo Zhang<sup>1</sup>    Laurent Perrussel<sup>2</sup>

<sup>1</sup>Intelligent Systems Laboratory  
University of Western Sydney, Australia

<sup>2</sup>IRIT, University of Toulouse, France

IJCAI-11

# Contributions



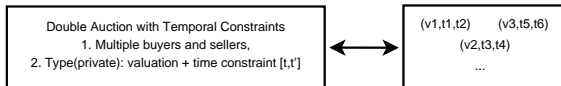
# Outline

- 1 The Model
  - The Domain
  - The Question
  - A Solution
- 2 Augmentation-based Mechanism
  - Graphical Representation
  - Allocation Policy Design
  - Payment Policy Design
- 3 Conclusion

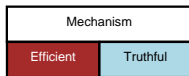
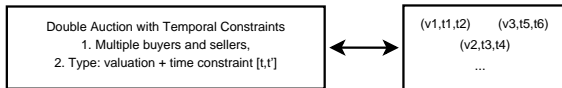
# Outline

- 1 The Model
  - The Domain
  - The Question
  - A Solution
- 2 Augmentation-based Mechanism
- 3 Conclusion

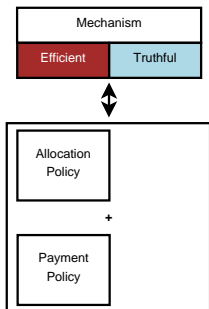
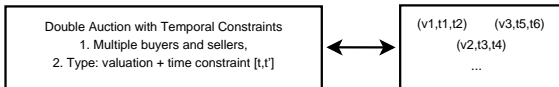
# Double Auction with Temporal Constraints



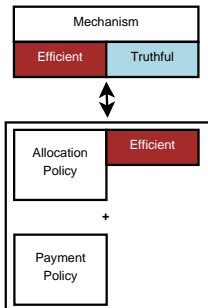
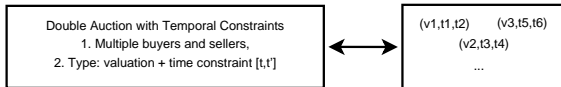
# Efficient & Truthful Mechanism



# Components of the Mechanism

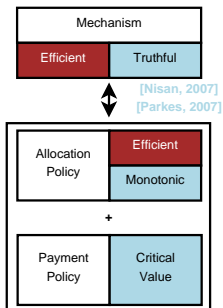
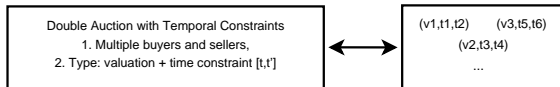


# Achieving Efficiency

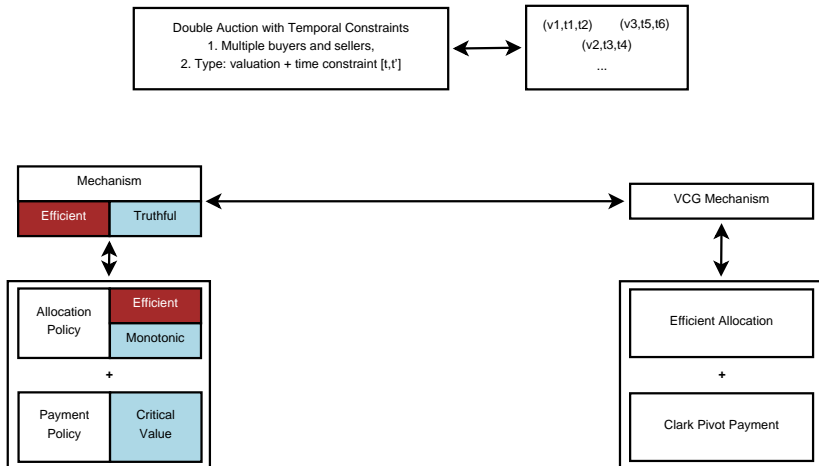




# Achieving Truthfulness



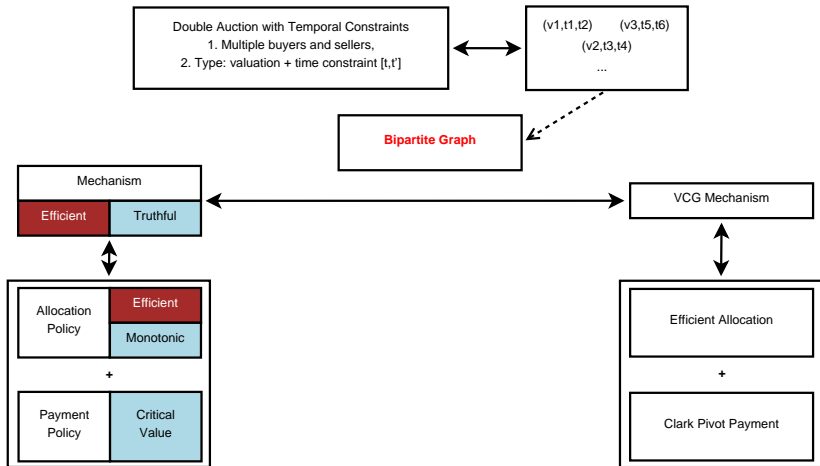
# The VCG Mechanism



# Outline

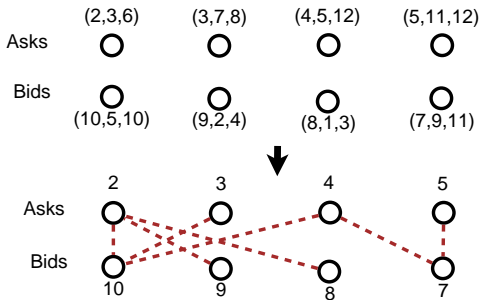
- 1 The Model
- 2 Augmentation-based Mechanism**
  - Graphical Representation
  - Allocation Policy Design
  - Payment Policy Design
- 3 Conclusion

# Constructing the Graph



# Constructing the Graph (an Example)

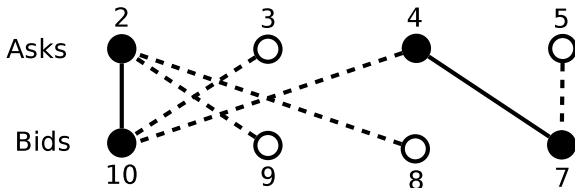
- An ask  $\theta_i = (v_i, s_i, e_i)$  and a bid  $\theta_j = (v_j, s_j, e_j)$  are **matchable** (i.e. item exchanging can happen between  $i$  and  $j$ ) iff  $v_i \leq v_j$  and  $[s_i, e_i] \cap [s_j, e_j] \neq \emptyset$ .



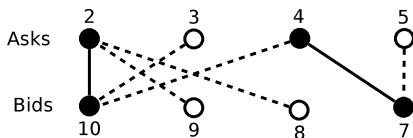
# Matching & Alternating Paths

## Definition

Given a graph  $G$ , a **matching**  $M$  in  $G$  is a set of pairwise non-adjacent edges, i.e. no two edges share a common vertex.



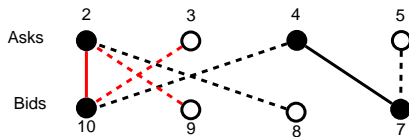
# Matching & Alternating Paths



Given a matching  $M$ ,

- an  **$M$ -alternating path** is a path in which the edges belong alternatively to  $M$  and not to  $M$ .

# Matching & Alternating Paths

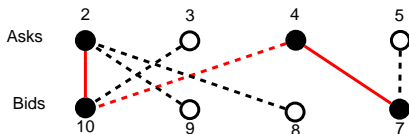


Given a matching  $M$ ,

- an  **$M$ -augmenting path** is an  $M$ -alternating path whose endpoints are free/unmatched. (3,10,2,9)



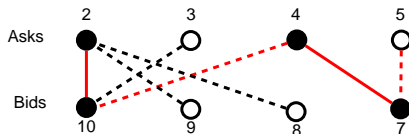
# Matching & Alternating Paths



Given a matching  $M$ ,

- an  **$M$ -abridging path** is an  $M$ -alternating path whose first edge and last edge are in  $M$ . **(2,10,4,7)**

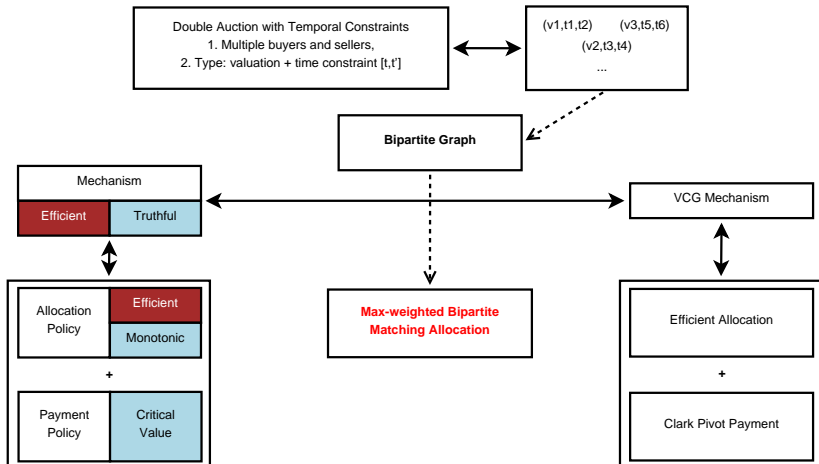
# Matching & Alternating Paths



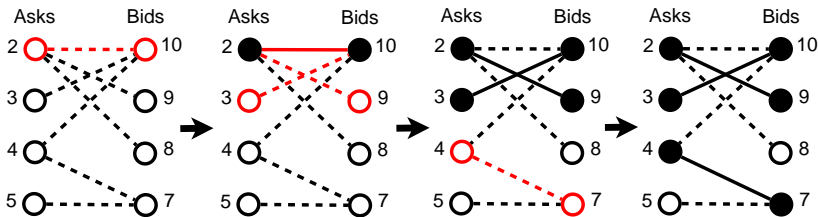
Given a matching  $M$ ,

- an  **$M$ -replacement path** is an  $M$ -alternating path where one of the endpoints is free and one of the ending edges is in  $M$ . **(2,10,4,7,5)**

# Maximum-weighted Bipartite Matching Allocation



# Maximum-weighted Bipartite Matching



# The Allocation Policy

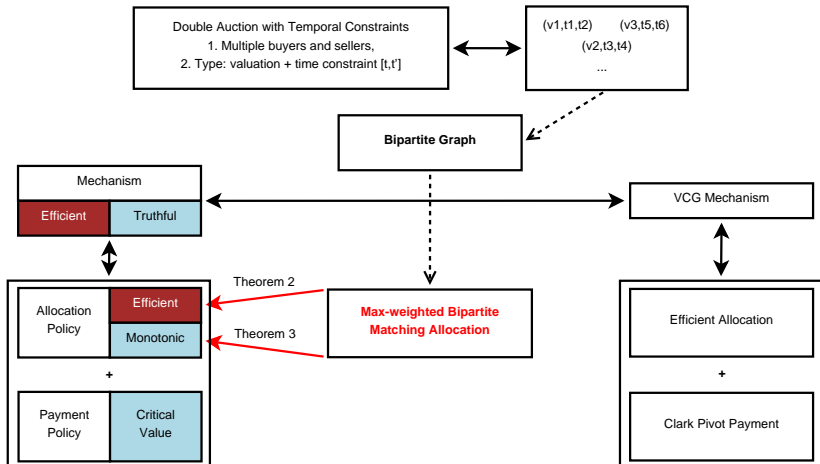
## Initialization:

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

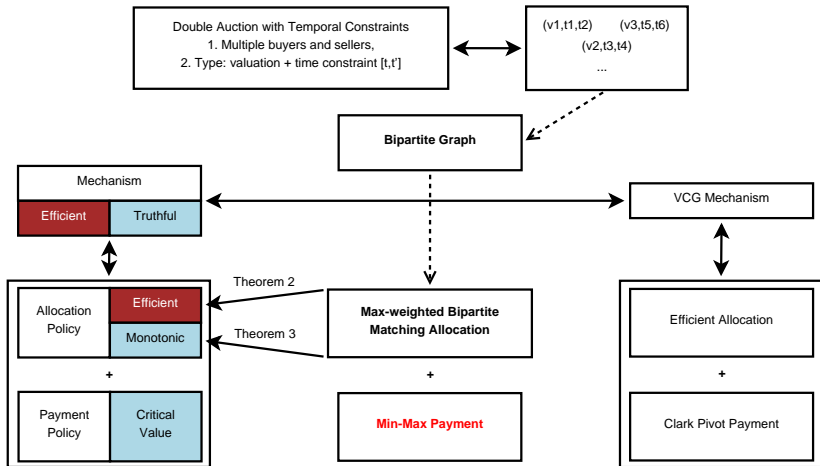
## Recursion:

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

# Properties of The Allocation



# Min-Max Payment



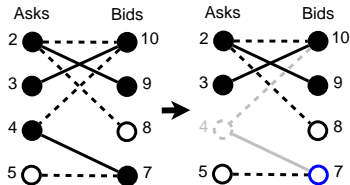
# The Intuition

For each matched ask/bid, looking for **the best substitution** and **the lowest loss** if the ask/bid was not participated.

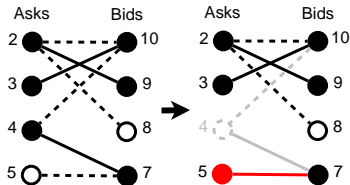


# For an matched ask

- each **abridging path**, *starting from the ask*, gives a way to remove the ask by unmatching another bid,
- each **replacement path**, *starting from the ask*, gives a way to remove the ask by giving a substitution.



the lowest loss



the best substitution

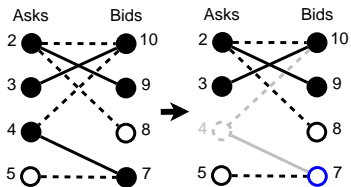
# The Payment Policy

$$x_i(\theta) = \begin{cases} \min_{p \in A \cup R} v(\text{ending}(p)), & \text{if } i \text{ is seller} \\ \max_{p \in A \cup R} v(\text{ending}(p)), & \text{if } i \text{ is buyer} \end{cases}$$

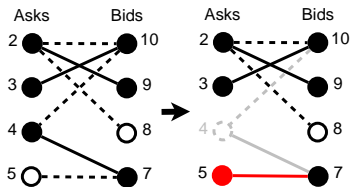
where

- $A$  is a set of all **abridging** paths starting from  $\theta_i$ ,
- $R$  is a set of all **replacement** paths starting from  $\theta_i$ ,
- and  $v(\text{ending}(p))$  is the valuation of the ending vertex, the endpoint other than  $\theta_i$ , of path  $p$ .

# For an matched ask



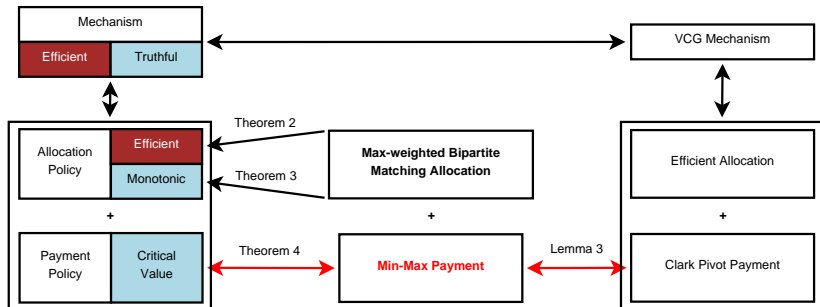
the lowest loss



the best substitution



# Properties of the Payment Policy



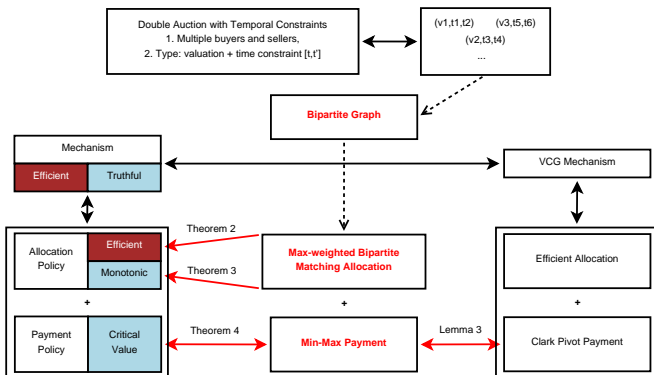
- does **NOT** need to rerun the allocation for each matched ask/bid,
- can be implemented  $O(n)$  times **faster** than Clarke pivot payment.

# Outline

- 1 The Model
- 2 Augmentation-based Mechanism
- 3 Conclusion**

## An efficient and truthful (VCG) double auction with time constraints:

- graphical representation of constraints
- augmentation-based policies
- a faster algorithm for computing VCG payment (critical value)



# Thank you for your attention!

