Mechanism Design for Double Auctions with Temporal Constraints

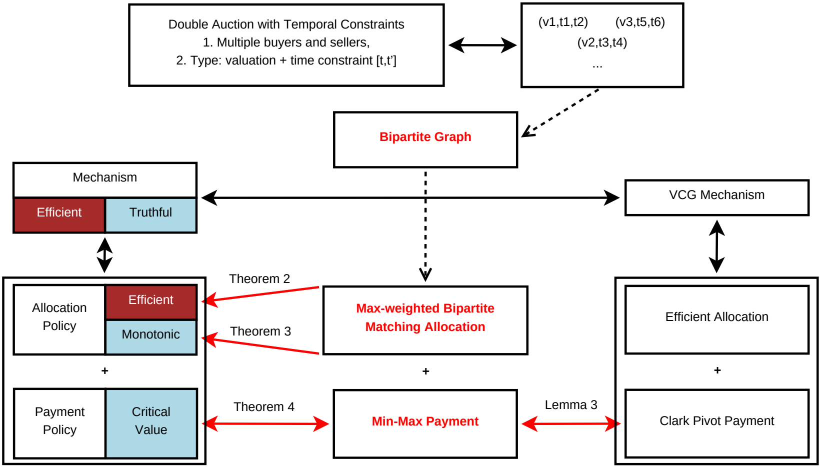
Dengji Zhao[1,2]
Dongmo Zhang[1] Laurent Perrussel[2]

[1]Intelligent Systems Laboratory
University of Western Sydney, Australia

[2]IRIT, University of Toulouse, France

IJCAI-11

Contributions



Outline

Outline

Double Auction with Temporal Constraints

Double Auction with Temporal Constraints
1. Multiple buyers and sellers,
2. Type(private): valuation + time constraint [t,t']

(v1,t1,t2) (v3,t5,t6)
(v2,t3,t4)
...

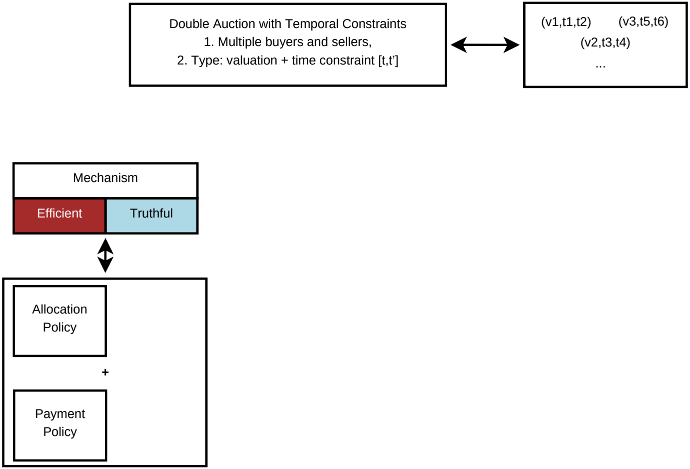
Efficient & Truthful Mechanism

Double Auction with Temporal Constraints
1. Multiple buyers and sellers,
2. Type: valuation + time constraint [t,t']

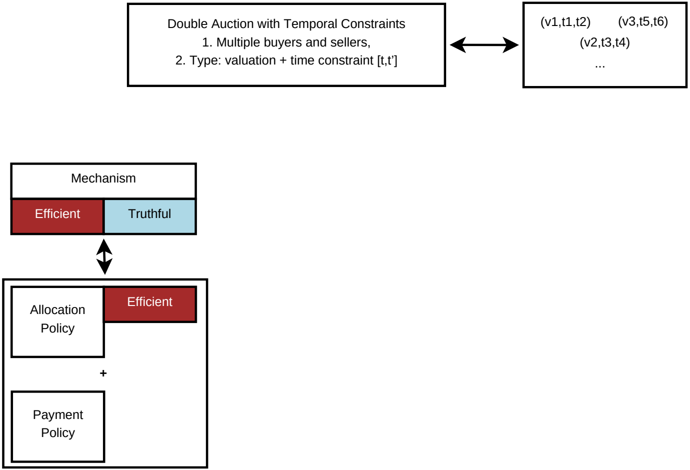
(v1,t1,t2) (v3,t5,t6)
(v2,t3,t4)
...

Mechanism	
Efficient	Truthful

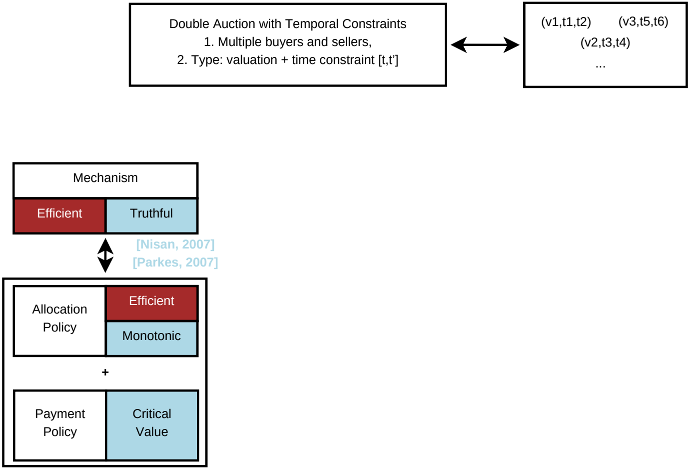
Components of the Mechanism



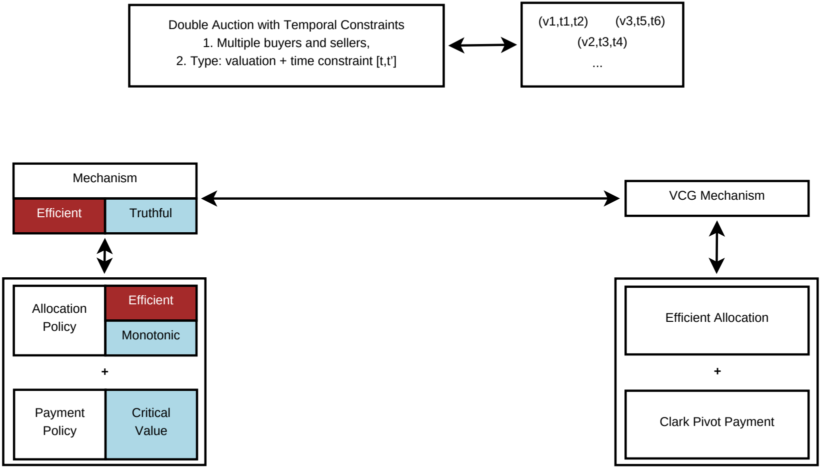
Achieving Efficiency



Achieving Truthfulness

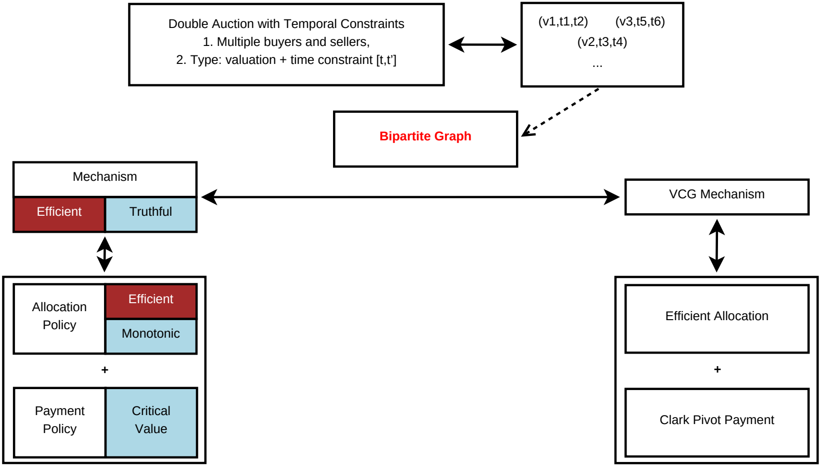


The VCG Mechanism



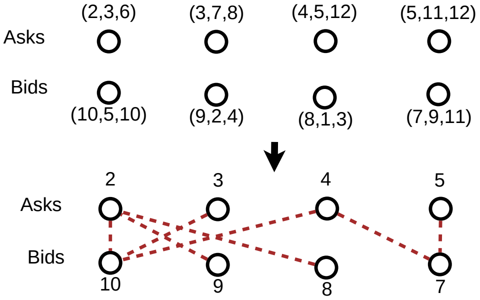
Outline

Constructing the Graph



Constructing the Graph (an Example)

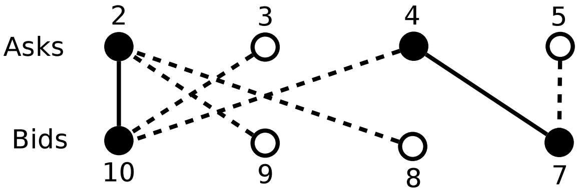
- An ask $\theta_i = (v_i, s_i, e_i)$ and a bid $\theta_j = (v_j, s_j, e_j)$ are matchable (i.e. item exchanging can happen between i and j) iff $v_i \leq v_j$ and $[\mathbf{s_i}, \mathbf{e_i}] \cap [\mathbf{s_j}, \mathbf{e_j}] \neq \varnothing$.



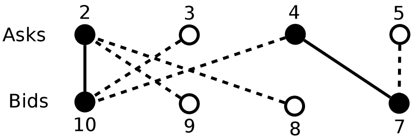
Matching & Alternating Paths

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.



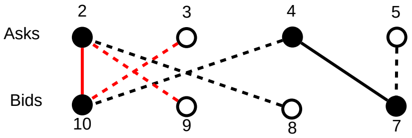
Matching & Alternating Paths



Given a matching M,

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

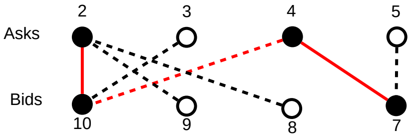
Matching & Alternating Paths



Given a matching M,

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

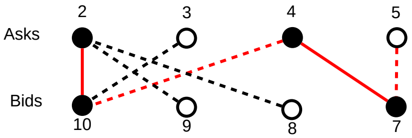
Matching & Alternating Paths



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)

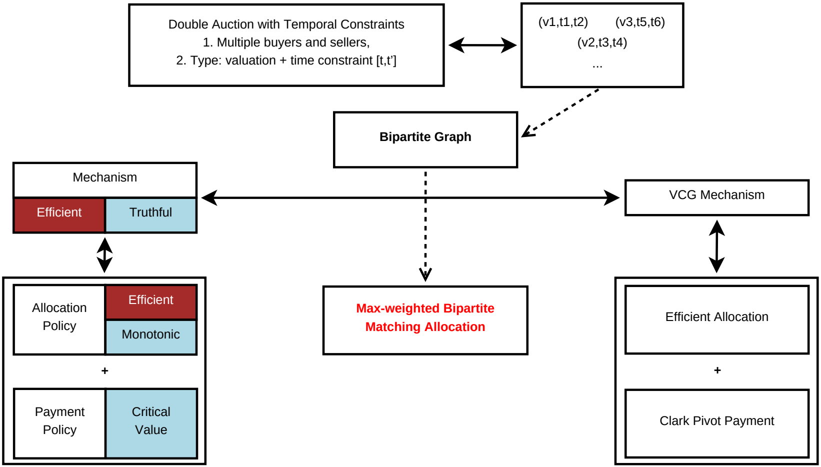
Matching & Alternating Paths



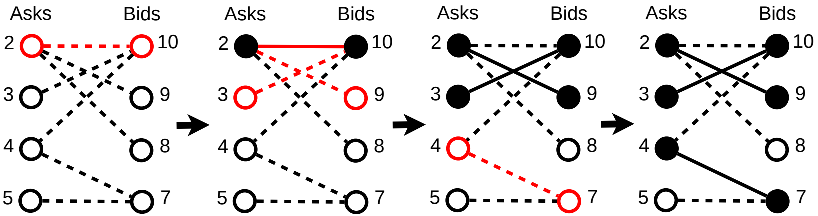
Given a matching M,

- 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

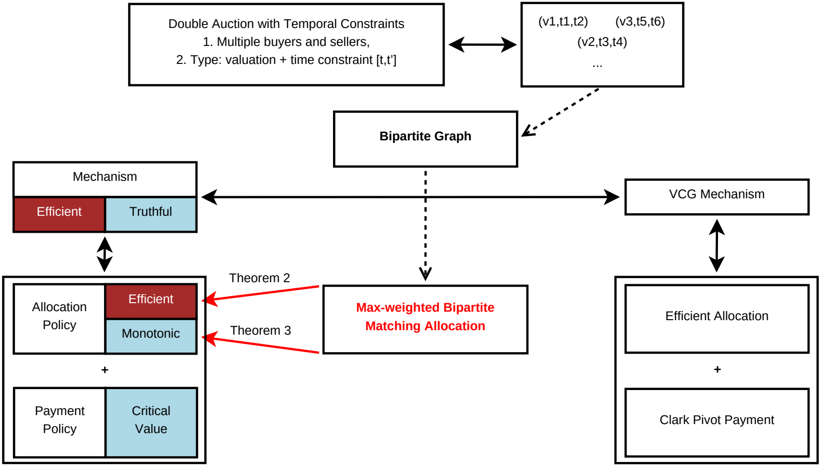
Initialization:

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

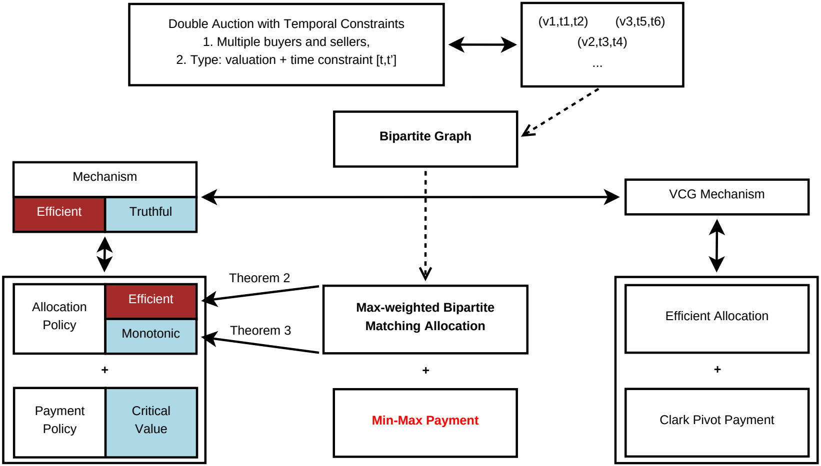
Recursion:

- $AugPath = \{p : \Delta(p) > 0$ 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 $M = M \oplus \hat{p}$.

Properties of The Allocation



Min-Max Payment

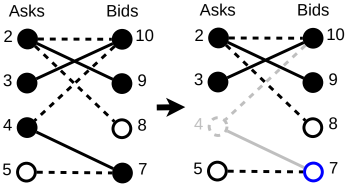


The Intuition

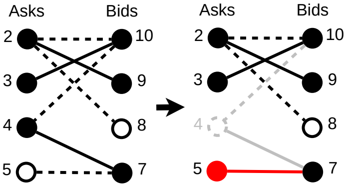
For each matched ask/bid, looking for **the best substitution** and **the lowest loss** if the ask/bid was not participated.

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

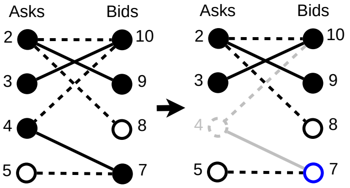
The Payment Policy

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

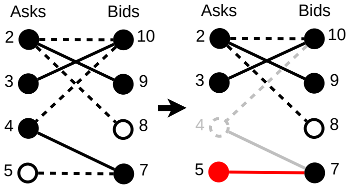
where

- A 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 an matched ask

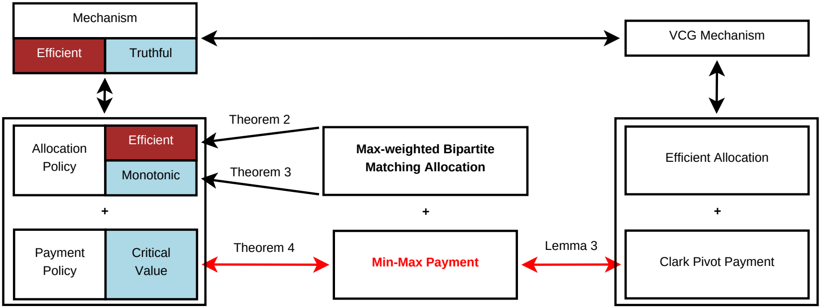


the lowest loss

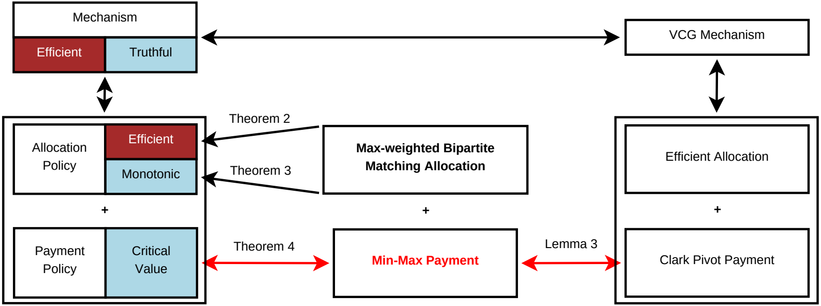


the best substitution

Properties of the Payment Policy



Properties of the Payment Policy

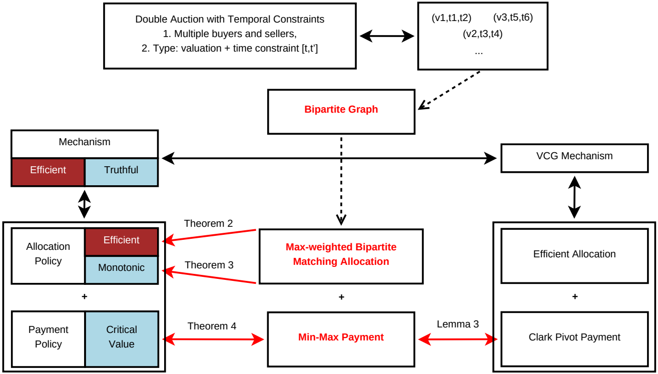


- does NOT need to rerun the allocation for each matched ask/bid,
- can be implemented $O(n)$ times faster than Clarke pivot payment.

Outline

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!