A Theory of House Allocation and Exchange Mechanisms

Marek Pycia
_Penn State_

M. Utku Ünver
_University of Pittsburgh_

June 19, 2008
Allocation and Exchange of Indivisible Goods without Monetary Transfers

- Exchange of “houses” (Shapley and Scarf, 1974)
Allocation and Exchange of Indivisible Goods without Monetary Transfers

- Exchange of “houses” (Shapley and Scarf, 1974)
- Allocation of dormitory rooms (Abdulkadiroğlu and Sönmez, 1999)
Allocation and Exchange of Indivisible Goods without Monetary Transfers

- Exchange of “houses” (Shapley and Scarf, 1974)
- Allocation of dormitory rooms (Abdulkadiroğlu and Sönmez, 1999)
- Exchange of live donor kidneys (Roth, Sönmez, and Ünver, 2004)
Allocation and Exchange of Indivisible Goods without Monetary Transfers

- Exchange of “houses” (Shapley and Scarf, 1974)
- Allocation of dormitory rooms (Abdulkadiroğlu and Sönmez, 1999)
- Exchange of live donor kidneys (Roth, Sönmez, and Ünver, 2004)
- Allocation of tasks
Satisfactory Mechanisms for Revelation of Preferences

- Pareto-efficient
Satisfactory Mechanisms for Revelation of Preferences

- Pareto-efficient
- Coalitionally strategy-proof
Constructions of House Allocation and Exchange Mechanisms

- Shapley and Scarf (1974) (David Gale’s top trading cycles)
Constructions of House Allocation and Exchange Mechanisms

- Shapley and Scarf (1974) (David Gale’s top trading cycles)
Constructions of House Allocation and Exchange Mechanisms

- Shapley and Scarf (1974) (David Gale’s top trading cycles)
- Abdulkadiroğlu and Sönmez (1999)
Constructions of House Allocation and Exchange Mechanisms

- Shapley and Scarf (1974) (David Gale’s top trading cycles)
- Abdulkadiroğlu and Sönmez (1999)
- Pápai (2000)
The Contribution of This Paper

We introduce the class of “trading cycles with brokers and owners” direct mechanisms and show that

- Each mechanism in the class is Pareto-efficient and coalitionally strategy-proof

Pycia & Ünver ()

House Allocation & Exchange

June 19, 2008
We introduce the class of “trading cycles with brokers and owners” direct mechanisms and show that

- Each mechanism in the class is Pareto-efficient and coalitionally strategy-proof
- Each Pareto-efficient and coalitionally strategy-proof direct mechanism is in the class
We introduce the class of “trading cycles with brokers and owners” direct mechanisms and show that

- Each mechanism in the class is Pareto-efficient and coalitionally strategy-proof
- Each Pareto-efficient and coalitionally strategy-proof direct mechanism is in the class
- The new mechanisms can be used to solve practical design problems that are beyond the reach of the previously known mechanisms
A finite set of agents $I$ and a finite set of houses $H$, $\#H \geq \#I$
A finite set of agents $I$ and a finite set of houses $H$, $\#H \geq \#I$

Each agent $i$ has a strict preference relation $\succ_i$ over houses
A finite set of agents $I$ and a finite set of houses $H$, $\#H \geq \#I$

Each agent $i$ has a strict preference relation $\succ_i$ over houses

A **direct mechanism** $\varphi$ is a function from preference profiles into (final) allocations such that
Model of House Allocation

- A finite set of agents $I$ and a finite set of houses $H$, $\#H \geq \#I$
- Each agent $i$ has a strict preference relation $\succ_i$ over houses
- A **direct mechanism** $\varphi$ is a function from preference profiles into (final) allocations such that
  - each agent gets one house, and
Model of House Allocation

- A finite set of agents $I$ and a finite set of houses $H$, $\#H \geq \#I$
- Each agent $i$ has a strict preference relation $\succ_i$ over houses
- A **direct mechanism** $\varphi$ is a function from preference profiles into (final) allocations such that
  - each agent gets one house, and
  - each house is allocated to at most one agent
Top-Trading-Cycles (TTC) Algorithm

In each round some agents and houses are matched and removed from the problem.
Top-Trading-Cycles (TTC) Algorithm

- In each round some agents and houses are matched and removed from the problem.
- The algorithm terminates when all agents are matched
At the beginning of the round, each unremoved house is controlled by an unremoved agent.
A Round of the TTC Algorithm

- At the beginning of the round, each unremoved house is **controlled** by an unremoved agent.
- Each house points to the agent that controls it.
A Round of the TTC Algorithm

- At the beginning of the round, each unremoved house is controlled by an unremoved agent.
- Each house points to the agent that controls it.
- Each agent points to his most preferred unremoved house.

In the resultant directed graph, there exists at least one exchange cycle in which:

\[ i_1 \] points to the house of \( i_2 \),
\[ i_2 \] points to the house of \( i_3 \),
\[ \ldots \]
\[ i_k \] points to the house of \( i_1 \).

Agents and houses in each exchange cycle are matched and removed.
A Round of the TTC Algorithm

- At the beginning of the round, each unremoved house is controlled by an unremoved agent.
- Each house points to the agent that controls it.
- Each agent points to his most preferred unremoved house.
- In the resultant directed graph, there exists at least one exchange cycle in which
  \( i_1 \) points to the house of \( i_2 \), \( i_2 \) points to the house of \( i_3 \), ...
  \( i_k \) points to the house of \( i_1 \).
A Round of the TTC Algorithm

- At the beginning of the round, each unremoved house is controlled by an unremoved agent.
- Each house points to the agent that controls it.
- Each agent points to his most preferred unremoved house.
- In the resultant directed graph, there exists at least one exchange cycle in which
  \( i_1 \) points to the house of \( i_2 \), \( i_2 \) points to the house of \( i_3 \), ...
  \( i_k \) points to the house of \( i_1 \).
- Agents and houses in each exchange cycle are matched and removed.
At each round, each unremoved house is controlled by an unremoved agent, either as an
At each round, each unremoved house is controlled by an unremoved agent, either as an

owner or
At each round, each unremoved house is controlled by an unremoved agent, either as an

- owner or
- broker
At each round, each unremoved house is controlled by an unremoved agent, either as an owner or broker.

At any round, there is at most one broker and one brokered house.
At each round, each unremoved house is controlled by an unremoved agent, either as an owner or broker.

At any round, there is at most one broker and one brokered house.

The last remaining agent is an owner.
At each round, each unremoved house is controlled by an unremoved agent, either as an

- **owner** or
- **broker**

At any round, there is at most one broker and one brokered house

The last remaining agent is an owner

Only unremoved agents have control right
Each unremoved house points to the agent that controls it
TCBO Continued
A Round of the TCBO Algorithm

- Each unremoved house points to the agent that controls it
- The broker (if there is one) points to his most preferred unremoved house other than the brokered house
Each unremoved house points to the agent that controls it.

- The broker (if there is one) points to his most preferred unremoved house other than *the brokered house*.

- Each other unremoved agent points to his most preferred unremoved house.
Each unremoved house points to the agent that controls it

The broker (if there is one) points to his most preferred unremoved house other than the brokered house

Each other unremoved agent points to his most preferred unremoved house

In the resultant directed graph, there exists at least one exchange cycle
Each unremoved house points to the agent that controls it.
The broker (if there is one) points to his most preferred unremoved house other than the brokered house.
Each other unremoved agent points to his most preferred unremoved house.
In the resultant directed graph, there exists at least one exchange cycle.
Agents and houses in each exchange cycle are matched and removed.
TTC Algorithms are a Proper Subclass of TCBO Algorithms

- Each TTC algorithm is a TCBO algorithm with no brokers
Example of TCBO
The Structure of Control Rights

Three agents $i_1, i_2, i_3$ and three houses $h_1, h_2, h_3$
The structure of control rights

\[
\begin{array}{c|c|c}
   h_1 & h_2 & h_3 \\
   \hline
   i_1 \text{ (broker)} & i_2 & i_3 \\
   i_3 & i_2 & i_1 \\
   i_1 & i_1 & i_1 \\
\end{array}
\]
Three agents $i_1, i_2, i_3$ and three houses $h_1, h_2, h_3$

The structure of control rights

<table>
<thead>
<tr>
<th>$h_1$</th>
<th>$h_2$</th>
<th>$h_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_1$</td>
<td>$i_2$</td>
<td>$i_3$</td>
</tr>
<tr>
<td>$i_3$</td>
<td>$i_2$</td>
<td>$i_1$</td>
</tr>
<tr>
<td>$i_1$</td>
<td>$i_1$</td>
<td>$i_1$</td>
</tr>
</tbody>
</table>

Preferences

$h_1 \succ_i h_2 \succ_i h_3$

$h_1 \succ_i h_2 \succ_i h_3$

$h_1 \succ_i h_3 \succ_i h_2$
Round 1:
i_1\text{ brokers } h_1,\ i_2\text{ owns } h_2,\ i_3\text{ owns } h_3
There exists a cycle
\[ h_1 \rightarrow i_1 \rightarrow h_2 \rightarrow i_2 \rightarrow h_1 \]
Pairs \((i_1, h_2), (i_2, h_1)\) are matched and removed.
Round 2:

\((i_1, h_2), (i_2, h_1)\) are already matched and removed; \(i_3\) owns \(h_3\)

There exists a cycle

\[ h_3 \rightarrow i_3 \rightarrow h_3 \]

The pairs \((i_3, h_3)\) is matched and removed. The final allocation is

\[ \{(i_1, h_2), (i_2, h_1), (i_3, h_3)\} \].
For any matching \( \sigma \) of removed agents and houses, the control rights structure satisfies:

- owners persist
- there exists at most one broker and one brokered house after \( \sigma \)
- brokers persist (two exceptions withstanding)
- brokers do not own any house
Let $\sigma' = \sigma \cup \{(j, g)\} \supsetneq \sigma$; if $k$ brokers $e$ at $\sigma$, then either

- at least one agent owns a house both at $\sigma$ and $\sigma'$, and $k$ brokers $e$ at $\sigma'$, or
- no agent owns a house both at $\sigma$ and $\sigma'$, or
- exactly one agent $i$ owns a house both at $\sigma$ and $\sigma'$, and
  (i) agent $i$ owns $e$ at $\sigma'$, and
  (ii) at every $\sigma'' \supset \sigma' \cup \{(i, e)\}$ at which $k$ is unmatched, $k$ owns all houses that $i$ owns at $\sigma$. 

Pycia & Ünver ()
House Allocation & Exchange
June 19, 2008 17 / 23
The TCBO mechanisms are the mechanisms implemented by the TCBO algorithm with control rights structures satisfying conditions (1)-(4)

- All TTC-based mechanisms are proper subclasses of TCBO mechanisms: Examples
The TCBO mechanisms are the mechanisms implemented by the TCBO algorithm with control rights structures satisfying conditions (1)-(4)

- All TTC-based mechanisms are proper subclasses of TCBO mechanisms: Examples
  - Serial Dictatorship (Svensson, 1994)
The TCBO mechanisms are the mechanisms implemented by the TCBO algorithm with control rights structures satisfying conditions (1)-(4)

- All TTC-based mechanisms are proper subclasses of TCBO mechanisms: Examples
  - Serial Dictatorship (Svensson, 1994)
The TCBO mechanisms are the mechanisms implemented by the TCBO algorithm with control rights structures satisfying conditions (1)-(4)

- All TTC-based mechanisms are proper subclasses of TCBO mechanisms: Examples
  - Serial Dictatorship (Svensson, 1994)
  - Core Mechanism (Shapley and Scarf, 1974)
The TCBO mechanisms are the mechanisms implemented by the TCBO algorithm with control rights structures satisfying conditions (1)-(4)

- All TTC-based mechanisms are proper subclasses of TCBO mechanisms: Examples
  - Serial Dictatorship (Svensson, 1994)
  - Core Mechanism (Shapley and Scarf, 1974)
  - Hierarchical Exchange (Papai, 2000)
Theorem.

1 Each TCBO mechanisms is Pareto-efficient and coalitionally strategy-proof
Main Result

Theorem.

1. Each TCBO mechanisms is Pareto-efficient and coalitionally strategy-proof.
2. Each Pareto-efficient and coalitionally strategy-proof direct mechanism is TCBO.
Manager assigns $n$ tasks $t_1, \ldots, t_n$ to $n$ employees $w_1, \ldots, w_n$.
Manager assigns $n$ tasks $t_1, \ldots, t_n$ to $n$ employees $w_1, \ldots, w_n$

Manager wants the allocation to be Pareto-efficient with regard to the employees’ preferences
Manager assigns $n$ tasks $t_1, \ldots, t_n$ to $n$ employees $w_1, \ldots, w_n$
Manager wants the allocation to be Pareto-efficient with regard to the employees’ preferences
Within this constraint, she would like to avoid assigning task $t_1$ to employee $w_1$. 
Manager assigns $n$ tasks $t_1, ..., t_n$ to $n$ employees $w_1, ..., w_n$

Manager wants the allocation to be Pareto-efficient with regard to the employees’ preferences

Within this constraint, she would like to avoid assigning task $t_1$ to employee $w_1$.

Not knowing employees’ preferences, she wants to use a coalitionally strategy-proof direct mechanism.
Using a TCBO, the manager can achieve her objective without the extreme discrimination of employee \( w_1 \), as follows

1. Make \( w_1 \) the broker of \( t_1 \)
Using a TCBO, the manager can achieve her objective without the extreme discrimination of employee $w_1$, as follows

1. Make $w_1$ the broker of $t_1$
2. Endow employees $w_2, \ldots, w_n$ with ownership rights over tasks $t_2, \ldots, t_n$ (for instance, make $w_i$ the owner of $t_i$)
Using a TCBO, the manager can achieve her objective without the extreme discrimination of employee $w_1$, as follows

1. Make $w_1$ the broker of $t_1$
2. Endow employees $w_2, ..., w_n$ with ownership rights over tasks $t_2, ..., t_n$ (for instance, make $w_i$ the owner of $t_i$)
3. run TCBO
Conclusion

We introduced the class of “trading cycles with brokers and owners” direct mechanisms and showed that

- Each mechanism in the class is Pareto-efficient and coalitionally strategy-proof
We introduced the class of “trading cycles with brokers and owners” direct mechanisms and showed that

- Each mechanism in the class is Pareto-efficient and coalitionally strategy-proof
- Each Pareto-efficient and coalitionally strategy-proof direct mechanism is in the class
Conclusion

We introduced the class of “trading cycles with brokers and owners” direct mechanisms and showed that

- Each mechanism in the class is Pareto-efficient and coalitionally strategy-proof
- Each Pareto-efficient and coalitionally strategy-proof direct mechanism is in the class
- The new mechanisms can be used to solve practical design problems that are beyond the reach of the previously known mechanisms.
Conclusion

We introduced the class of “trading cycles with brokers and owners” direct mechanisms and showed that

- Each mechanism in the class is Pareto-efficient and coalitionally strategy-proof
- Each Pareto-efficient and coalitionally strategy-proof direct mechanism is in the class
- The new mechanisms can be used to solve practical design problems that are beyond the reach of the previously known mechanisms.
- Characterization result can be extended to problem domains with initial property rights such as on-campus housing, kidney exchange.
Satisfactory Mechanisms in Other Contexts
