Networks II: Market Design—Lecture 10
Markets with Initial Endowments: The Core

ARPITA GHOSH

Dept. Of Information Science,
Cornell University

Recap: Pareto-efficiency, individual rationality and the core

- Every core matching is individually rational (IR) and Pareto-efficient (PE)
  - Consider one-person "coalition" $B = \{a\}$ and grand coalition $B = A$ respectively
- Is every IR matching in the core?
  - No: Simple 2-agent, 2-item example where swapping improves allocation
- Is every PE matching in the core?
  - No, again: A Pareto-efficient matching need not be individually rational!
- What if a matching is both PE and IR?
  - Not enough: Matching can be both PE and IR but not in the core!

Existence

- A matching M is in the **core** if there is no coalition of agents $B \subseteq A$, and a matching $\hat{M}$, such that
  - For any $a \in B$, $\hat{M}(a)$ is the *initial house* of some $b \in B$, and
  - $\hat{M}(a) \succeq_a M(a)$ for all $a \in B$ and $\hat{M}(b) \succ_b M(b)$ for some $b \in B$
- Does such a matching always exist?

Theorem (Shapley and Scarf 1974)

Assume agents have strict preferences. There exists a core matching for any housing market.

Proof by construction(!): *Top Trading Cycles (TTC) algorithm*

Looking for a core matching

An example: Initial endowments a-x, b-y, c-z

(A matching M is in the **core** if there is no coalition of agents $B \subseteq A$, and a matching $\hat{M}$, such that

- For any $a \in B$, $\hat{M}(a)$ is the *initial house* of some $b \in B$, and
- $\hat{M}(a) \succeq_a M(a)$ for all $a \in B$ and $\hat{M}(b) \succ_b M(b)$ for some $b \in B$)

- Preferences are
  a: x $\succ$ y $\succ$ z, b: x $\succ$ y $\succ$ z, c: y $\succ$ z $\succ$ x

- Is there a core matching in this example?
  - Yes: a-x, b-y, c-z
  - How would you argue this is in the core?

Another example: Initial endowments a-x, b-y, c-z

(A matching M is in the **core** if there is no coalition of agents $B \subseteq A$, and a matching $\hat{M}$, such that

- For any $a \in B$, $\hat{M}(a)$ is the *initial house* of some $b \in B$, and
- $\hat{M}(a) \succeq_a M(a)$ for all $a \in B$ and $\hat{M}(b) \succ_b M(b)$ for some $b \in B$)

- Preferences are
  a: z $\succ$ y $\succ$ x, b: z $\succ$ x $\succ$ y, c: x $\succ$ y $\succ$ z

- Is there a core matching in this market?
  - Yes again: a-z, b-y, c-x!
  - How would you argue this is in the core?

Gale's Top Trading Cycles (TTC) algorithm: Preliminaries

- Preferences are strict (also assume complete, for simplicity)
- Setting up the algorithm:
  - $p_k = (a_k, h_k)$: Pair of agent k and her initial endowment h_k
  - P: (Running) set of pairs p_k
  - Directed graph $G(P)$: Directed edge from $p_i = (a_i, h_i) \in P$ to $p_j = (a_j, h_j) \in P$ if item h_j is agent a_i's top preference amongst items in P
- Lemma: If P is non-empty, there exists at least one cycle in the graph $G(P)$, **and** no cycles intersect.

Top Trading Cycles (TTC) algorithm

- Initialize P to be set of all agent-item pairs in the market
- Repeat until P is empty
  - Create graph $G(P)$: Each agent points to (pair corresponding to) her most-preferred item in P
  - By Lemma, there exists at least one cycle in $G(P)$ **and** no cycles intersect
  - Assign each agent in a cycle the item she is pointing to
  - Update P to remove (pairs p_k corresponding to) all cycles in $G(P)$ from P
- Basic facts: This procedure
  (i) terminates in finite steps, and
  (ii) produces a matching

Proving that TTC returns a core matching

- Let M_T be matching returned by TTC
- **If** M_T is not in the core: There exists coalition $C^* \subseteq A$ that 'prefers' a different matching $\hat{M}$
- C_{strict}: Subset of agents in C^* who *strictly* prefer their allocation under $\hat{M}$ to that in M_T
- Note that C_{strict}:
  - Is non-empty (Why?: Definition of improved matching $\hat{M}$)
  - Need not equal C^* (Why?: Recall C^* can contain agents who only get a weak improvement (example in earlier lecture))
- Let $a_0 \in C_{strict}$ be *an* agent who is matched **earliest** in 'TTC order', amongst all agents in C_{strict}
  - Let I be the step/round in which a_0 is matched in TTC

TTC returns a core matching

Continuing: **If** M_T is not in the core:

Fact 1. Let agent a_1 be the owner of item $\hat{M}(a_0)$. Then, $a_1 \in C^*$. Why?

- By definition of core: Owner of $\hat{M}(a_0)$ must be in C^*

Fact 2. Agent a_1 is matched by TTC algorithm in a strictly earlier 'step' $l' < l$ than agent a_0. Why?

- a_0 points to her most preferred item in $G(P)$ amongst remaining items at step l (and is assigned this item $M_T(a_0)$)
- Assumption: a_0 (strictly) prefers item $\hat{M}(a_0)$, owned by a_1, to $M_T(a_0)$
- So if a_1 had not been removed in an earlier step, a_1 would still be in P at step l: a_0 would have pointed to a_1's item in step l, instead of pointing to $M_T(a_0)$

TTC returns a core matching

Continuing: **If** M_T is not in the core:

Fact 3. $\hat{M}(a_1) = M_T(a_1)$. Why?

- $\hat{M}(a_1) \succeq M_T(a_1)$ by definition of deviating coalition C^*
- But a_0 is *earliest* agent (in TTC order) to strictly prefer her allocation in $\hat{M}$ to M_T
- Preferences are strict—so a_1's allocation must be the same in M_T and $\hat{M}$: $\hat{M}(a_1) = M_T(a_1)$

Fact 4. Let a_2 be the owner of $\hat{M}(a_1) = M_T(a_1)$. Then, $a_2 \in C^*$, **and** $a_2 \in C_l'$. Why?

- $a_2 \in C^*$: By definition of coalition C^*, since a_2 owns $\hat{M}(a_1)$ (recall definition of the core!)
- $a_2 \in C_{l'}$: By definition of TTC algorithm, since a_2 owns $M_T(a_1)$

TTC returns a core matching

Continuing: **If** M_T is not in the core:

Fact 5. ... and so on: Each agent a_i, $i \geq 1$ is assigned same item in both M_T and $\hat{M}$, belonging to, say, agent a_{i+1}

- Note: So a_{i+1} belongs to both C^* and $C_{l'}$

Fact 6. Finite cycle $C_{l'}$: Some agent $a^* \in C^* \cap C'_l$ must obtain a_1's item in TTC matching M_T (along cycle C'_l) **and** in matching $\hat{M}$

- Two different agents a_0 and $a^* \neq a_0$ are allocated same item in $\hat{M}$: $\hat{M}$ is not a matching!
- **Contradiction:** Matching M_T must belong to the core