Networks II: Market Design—Lecture 5 Matching Markets with Non-transferable Utilities: One-sided Preferences

ARPITA GHOSH

Dept. Of Information Science,
Cornell University

Logistics

- Any late joiners: Welcome!
    - Catch up on CMS: Lectures; general information handouts
        - Course information and policies; homework policies and guidelines
- Office hours: **Now on CMS** (plenty of TA office hours!)
- HW1 coming up: Due Thursday Feb 7th noon
    - *Electronic upload* (by you!) on CMS
    - Reminders:
        - HW late policy
        - Note: Screenshots as proof-of-time do not work!
        - **Academic integrity:** Do not cheat (duh), and in particular, acknowledge discussants at top of HW
        - See homework guidelines document on CMS

Recap: Last time

- Generalizing from binary to rank-order preferences: Market $(A, H, \succ)$
- Rank-order preferences $\succ$: What is a ‘good’ matching?
  - Not enough information to choose ‘best’ matching
  - Instead ask: Which are **bad** matchings?
  - ‘Good’ matchings: ‘Not-bad’ matchings
- Pareto-efficient matchings

Outline for today

- Mechanisms
- What are good mechanisms?
- The serial dictatorship mechanism

Checking in: Pareto efficiency

- Market $(A, H, \succ)$
  - Agents A = {a,b,c}; houses H = {x,y,z}
  - Preferences: a: x $\succ$ z $\succ$ y; b: y $\succ$ z $\succ$ x
  - *c's preferences are missing*

- Is matching M_1 = (a-x, b-z, c-y) Pareto-efficient?
  - A Yes
  - B No
  - C I can't decide

- Is matching M_2 = (a-z, b-x, c-y) Pareto-efficient?
  - A Yes
  - B No
  - C I can't decide

Checking in: Pareto efficiency

- Consider a market $(A, H, \succ)$, and two distinct matchings M and $\hat{M}$ in this market. If matching M is Pareto-efficient, it must Pareto-dominate matching $\hat{M}$.
  - A True
  - B False
- Can a market $(A, H, \succ)$ have two different Pareto-efficient matchings?
  - A Yes
  - B No
- Is every market $(A, H, \succ)$ certain to have at least one Pareto-efficient matching?
  - A Yes
  - B No
  - C Shouldn't there be a law against too many questions?

Rank-order preferences: What is a 'good' matching?

- Recall market: 3 agents A,B,C and 3 houses x,y,z
- Preferences are A: x $\succ$ y; B: x $\succ$ z; C: z $\succ$ x
- Three matchings: $M_2 =$ (A-x, B-z), $M_3 =$ (A-x, C-z), $M_4 =$ (A-y, B-x, C-z)
- Which of these is Pareto-efficient?
  - A M_2 only
  - B M_2 and M_3 only
  - C All of M_2, M_3, M_4
  - D M_4 only

Outline for today

- Mechanisms
- Pareto-efficiency and mechanisms
- The serial dictatorship mechanism

Allocating houses to agents: Mechanisms

- (Deterministic) mechanism: Rule that assigns a matching for each tuple $(A, H, \succ)$
  - Mechanism takes tuple $(A, H, \succ)$ as input
  - Returns *matching* M as output
- Mechanisms: Algorithms with input coming from *agents*
- Matchings and mechanisms are both *functions*: How are they different?
  - Matching: Function from (set of) *agents* to (set of)*objects*
  - Mechanism: *Function* from (set of) *preference profiles* $\succ = (\succ_a)_{a \in A}$ to (set of) *matchings*

Mechanisms: An example

- Mechanisms for housing markets: Tuple $(A, H, \succ)$
  - An example market: Three agents a,b,c; three houses x,y,z
  - Which of these adequately specifies a mechanism for this market?
    1. Return matching $a - x, b - z, c - y$ when input preference profile is a: x $\succ$ z $\succ$ y; b: x $\succ$ z $\succ$ y; c: x $\succ$ y $\succ$ z
    2. Return matching $a - x, b - z, c - y$ for all input preference profiles

    A Only (1)
    B Only (2)
    C Both (1) and (2)
    D Neither (1) nor (2)

- Mechanism: Rule to decide which matching M to choose given market $(A, H, \succ)$
  - Does not have to produce a 'good' matching to be a mechanism
  - But: *Has* to tell you 'what to do' for *any* possible input
- Recall: Matchings and mechanisms are both *functions*
  - Matching: Function from *agents* to *objects*
  - Mechanism: *Function* from (set of) preferences to (set of) matchings

Mechanisms: Another example

- Rewind for a moment: Binary $(0-1)$ preferences
  - Aside 1: Binary preferences as (weak) rank-order preferences
  - Aside 2: Term "mechanism" applies generally, beyond rank-order preferences!

- Is this a mechanism under binary preferences?
  (Assume some tie-breaking rule amongst matchings is specified)

1. Return a perfect matching if one exists
   - A Yes (*i.e.*, is a mechanism)
   - B No
2. Return a maximum matching
   - A Yes
   - B No

Outline for today

- Mechanisms
- Good mechanisms: Pareto-efficiency
- The serial dictatorship mechanism

Pareto-efficient mechanisms

- Mechanism: Assigns a matching for each preference profile
  - $\varphi(A, H, \succ)$: Matching returned by mechanism φ on input $(A, H, \succ)$
  - Notation: Sometimes use $\varphi(\succ)$ when (A, H) is clear from context
- So far, no requirements on goodness of matchings produced by mechanism: What are 'good' mechanisms?
  - Good mechanisms: Return 'good' matchings
- Mechanism φ is **Pareto-efficient** if $\varphi(\succ)$ is Pareto-efficient for every input preference profile $\succ$

Pareto-efficient mechanisms

- Pareto-efficient *matchings* versus Pareto-efficient *mechanisms*
- *Matching* M: Pareto-efficient if there is no other matching $\hat{M}$ that Pareto-dominates it
- *Mechanism* φ: Pareto-efficient if for every input tuple $(A, H, \succ)$, it returns a Pareto-efficient matching $M = \varphi(\succ)$ as output

An example

- Example: Agents a,b,c; houses x,y,z
- Preferences $\succ$ are a: x $\succ$ z $\succ$ y; b: x $\succ$ z $\succ$ y; c: x $\succ$y $\succ$ z
- $\varphi(\succ)$ returns matching a-x, b-z, c-y: Is φ Pareto-efficient?
  - A Yes
  - B No
  - C I don't know
  - D I don't like thinking
  - — Mechanism φ is Pareto-efficient if it returns a Pareto-efficient matching for *every* input: Need to know output for *all* inputs!
- $\varphi'(\succ)$ returns matching a-y, b-x, c-z: Is φ' Pareto-efficient?
  - A Yes
  - B No
  - C I don't know
  - D I *really* don't like thinking!

Outline for today

- Mechanisms
- Good mechanisms: Pareto-efficiency
- The serial dictatorship mechanism

A specific mechanism: Serial dictatorship

- Serial dictatorship, or *priority* mechanisms:
  - Specifies an order amongst agents
  - First agent receives her favorite good, second agent his favorite among remaining ones, ...
- Formally: Given a market $(A, H, \succ)$, serial dictatorship mechanisms specifies priority order f over A
  - $f(i)$: Agent with i^{th} priority
  - Agent $f(i)$ receives item ranked highest in her preference list amongst all items except those already taken by agents $f(1), \ldots, f(i-1)$

Serial dictatorship

- Serial dictatorship very easy to implement:
  - Decide order: Randomly, using existing priority order (e.g. seniority), ...
  - Let highest remaining applicant in order choose amongst remaining items
- Used in many applications (with variations):
  - NYC school choice system
  - Columbia and Harvard housing allocation (Pathak 2008, Kojima and Manea 2006, Che and Kojima 2008)
  - Office space allocation
  - ...
- Mechanism has many desirable properties: Perhaps related to wide use