

# Fair division

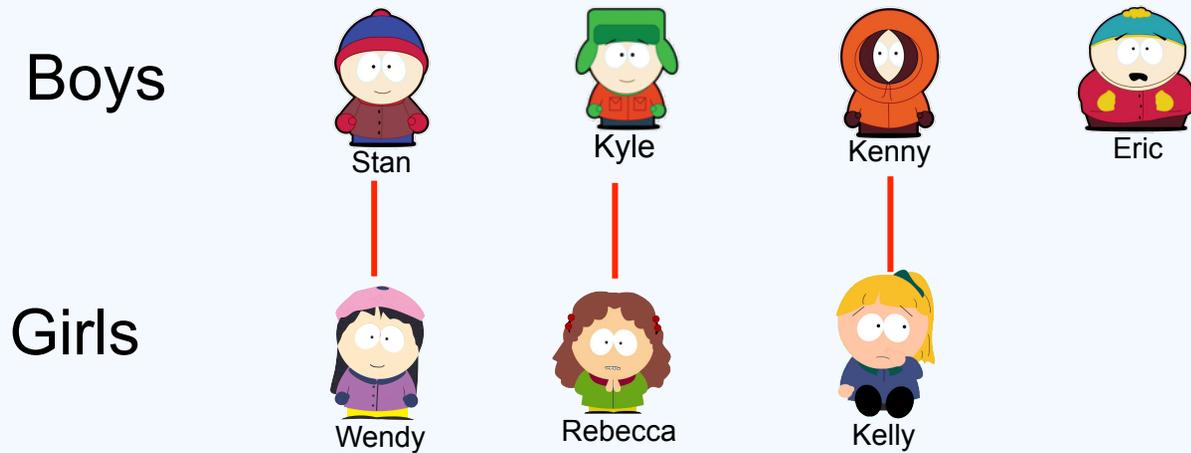
Lirong Xia



Rensselaer

Oct 7, 2013

# Last class: two-sided 1-1 stable matching



- Men-proposing deferred acceptance algorithm (DA)
  - outputs the men-optimal stable matching
  - runs in polynomial time
  - strategy-proof on men's side
- No matching mechanism is both stable and strategy-proof

# Today: FAIR division

- Fairness conditions
- Allocation of indivisible goods
  - serial dictatorship
  - Top trading cycle
- Allocation of divisible goods (cake cutting)
  - discrete procedures
  - continuous procedures

# Example 1

Agents



Houses



# Example 2

Agents

One divisible good



# Formal setting

- Agents  $A = \{1, \dots, n\}$
- Goods  $G$ : finite or infinite
- Preferences: represented by utility functions
  - agent  $j$ ,  $u_j: G \rightarrow \mathbb{R}$
- Outcomes = Allocations
  - $g: G \rightarrow A$
  - $g^{-1}: A \rightarrow 2^G$
- Difference with matching in the last class
  - 1-1 vs 1-many
  - Goods do not have preferences

# Efficiency criteria

- **Pareto dominance**: an allocation  $g$  Pareto dominates another allocation  $g'$ , if
  - all agents are not worse off under  $g'$
  - some agents are strictly better off
- **Pareto optimality**
  - allocations that are not Pareto dominated
- Maximizes social welfare
  - utilitarian
  - egalitarian

# Fairness criteria

- Given an allocation  $g$ , agent  $j_1$  **envies** agent  $j_2$  if  $u_{j_1}(g^{-1}(j_2)) > u_{j_1}(g^{-1}(j_1))$
- An allocation satisfies **envy-freeness**, if
  - no agent envies another agent
  - c.f. stable matching
- An allocation satisfies **proportionality**, if
  - for all  $j$ ,  $u_j(g^{-1}(j)) \geq u_j(G)/n$
- Envy-freeness implies proportionality
  - proportionality does not imply envy-freeness

# Why not...

- Consider fairness in other social choice problems
  - voting: does not apply
  - matching: when all agents have the same preferences
  - auction: satisfied by 2<sup>nd</sup> price auctions
- Use the agent-proposing DA in resource allocation (creating random rake preferences for the goods)
  - stability is no longer necessary
  - sometimes not 1-1
  - for 1-1 cases, other mechanisms may have better properties

# Allocation of indivisible goods

- House allocation
  - 1 agent 1 good
- Housing market
  - 1 agent 1 good
  - each agent originally owns a good
- 1 agent multiple goods (not discussed today)

# House allocation

- The same as two sided 1-1 matching except that the houses do not have preferences
- The serial dictatorship (SD) mechanism
  - given an order over the agents, w.l.o.g.  $a_1 \rightarrow \dots \rightarrow a_n$
  - in step  $j$ , let agent  $j$  choose her favorite good that is still available
  - can be either centralized or distributed
  - computation is easy

# Characterization of SD

- **Theorem.** Serial dictatorships are the only deterministic mechanisms that satisfy
  - strategy-proofness
  - Pareto optimality
  - neutrality
  - non-bossy
    - An agent cannot change the assignment selected by a mechanism by changing his report without changing his own assigned item
- Random serial dictatorship: paper presentation

# Why not agent-proposing DA

- Agent-proposing DA satisfies
  - strategy-proofness
  - Pareto optimality
- May fail neutrality



Stan

:  $h_1 > h_2$

$h_1: S > K$



Kyle

:  $h_1 > h_2$

$h_2: K > S$

- How about non-bossy?
  - No
- Agent-proposing DA when all goods have the same preferences = serial dictatorship

# Housing market

- Agent  $j$  initially owns  $h_j$
- Agents cannot misreport  $h_j$ , but can misreport her preferences
- A mechanism  $f$  satisfies **participation**
  - if no agent  $j$  prefers  $h_j$  to her currently assigned item
- An assignment is **in the core**
  - if no subset of agents can do better by trading the goods that they own amongst themselves
  - stronger than Pareto-optimality

# Example: core allocation



Stan

:  $h1 > h2 > h3$ , owns  $h3$



Kyle

:  $h3 > h2 > h1$ , owns  $h1$



Eric

:  $h3 > h1 > h2$ , owns  $h2$



Stan

:  $h2$



Kyle

:  $h3$



Eric

:  $h1$



Stan

:  $h1$



Kyle

:  $h3$



Eric

:  $h2$

Not in the core

In the core

# The top trading cycles (TTC) mechanism

- Start with: agent  $j$  owns  $h_j$
- In each round
  - built a graph where there is an edge from each available agent to the owner of her most-preferred house
  - identify all cycles; in each cycle, let the agent  $j$  gets the house of the next agent in the cycle; these will be their final allocation
  - remove all agents in these cycles

# Example

$a_1: h_2 > \dots$

$a_2: h_1 > \dots$

$a_3: h_4 > \dots$

$a_4: h_5 > \dots$

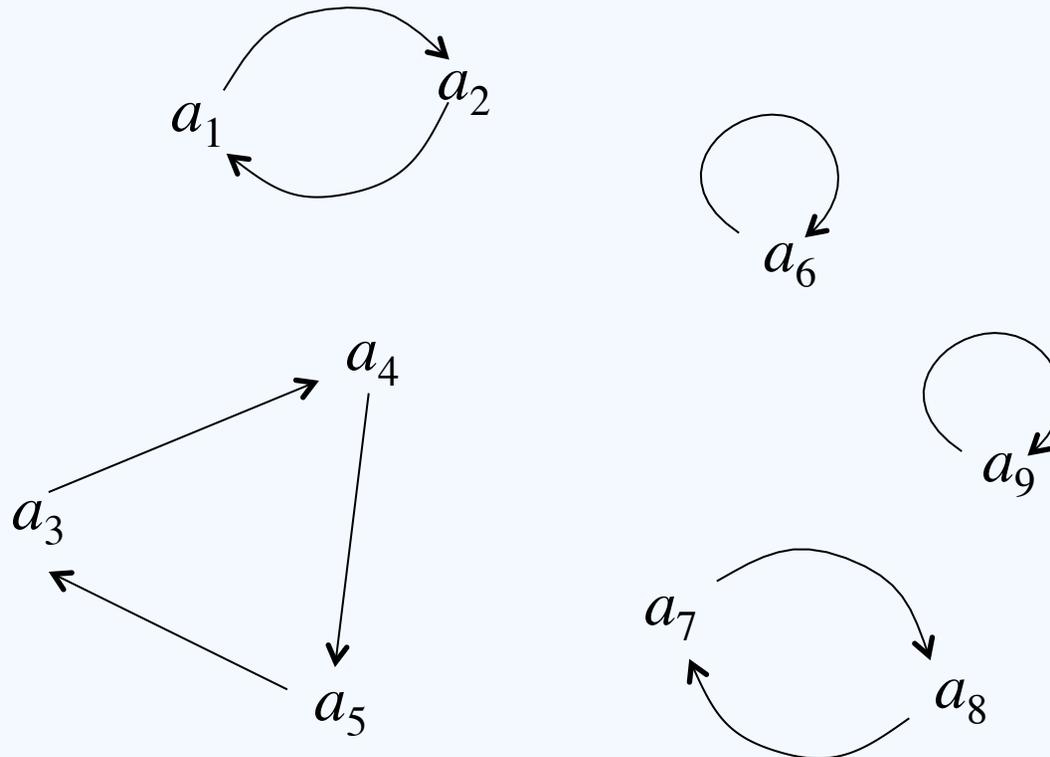
$a_5: h_3 > \dots$

$a_6: h_4 > h_3 > h_6 > \dots$

$a_7: h_4 > h_5 > h_6 > h_3 > h_8 > \dots$

$a_8: h_7 > \dots$

$a_9: h_6 > h_4 > h_7 > h_3 > h_9 > \dots$



# Properties of TTC

- **Theorem.** The TTC mechanism
  - is strategy-proof
  - is Pareto optimal
  - satisfies participation
  - selects an assignment in the core
    - the core has a unique assignment
  - can be computed in  $O(n^2)$  time
- Why not use TTC in 1-1 matching?
  - not stable
- Why not use TTC in house allocation (using random initial allocation)?
  - not neutral

# DA vs SD vs TTC

- All satisfy
  - strategy-proofness
  - Pareto optimality
  - easy-to-compute
- DA
  - stableness
- SD
  - neutrality
- TTC
  - chooses the core assignment

# Multi-issue resource allocation

- Each good is characterized by multiple issues
  - e.g. each presentation is characterized by topic and time
- Paper allocation
  - we have used SD to allocate the topic
  - we will use TTC for time
- Potential research project

# Allocation of one divisible good

- The set of goods is  $[0, 1]$       0 ————— 1
- Each utility function satisfies
  - Non-negativity:  $u_j(B) \geq 0$  for all  $B \subseteq [0, 1]$
  - Normalization:  $u_j(\emptyset) = 0$  and  $u_j([0, 1]) = 1$
  - Additivity:  $u_j(B \cup B') = u_j(B) + u_j(B')$  for disjoint  $B, B' \subseteq [0, 1]$
  - is continuous
- Also known as **cake cutting**
  - discrete mechanisms: as protocols
  - continuous mechanisms: use moving knives



# 2 agents: cut-and-choose

- Dates back to at least the Hebrew Bible [[Brams&Taylor, 1999, p. 53](#)]
- The cut-and-choose mechanism
  - 1<sup>st</sup> step: One player cuts the cake in two pieces (which she considers to be of equal value)
  - 2<sup>nd</sup> step: the other one chooses one of the pieces (the piece she prefers)
- Cut-and-choose satisfies
  - proportionality
  - envy-freeness
  - some operational criteria
    - each agent receive a continuous piece of cake
    - the number of cuts is minimum
    - is discrete

# More than 2 agents: The Banach-Knaster Last-Diminisher Procedure

- In each round
  - the first agent cut a piece
  - the piece is passed around other agents, who can
    - pass
    - cut more
  - the piece is given to the last agent who cut
- Properties
  - proportionality
  - **not envy-free**
  - the number of cut may not be minimum
  - is discrete

# The Dubins-Spanier Procedure

- A referee moves a knife slowly from left to right
- Any agent can say “stop”, cut off the piece and get it
- Properties
  - proportionality
  - **not envy-free**
  - minimum number of cuts (continuous pieces)
  - continuous mechanism

# Envy-free procedures

- $n = 2$ : cut-and-choose
- $n = 3$ 
  - The Selfridge-Conway Procedure
    - discrete, number of cuts is not minimum
  - The Stromquist Procedure
    - continuous, uses four simultaneous moving knives
- $n = 4$ 
  - no procedure produces continuous pieces is known
  - [Barbanel&Brams 04] uses a moving knife and may use up to 5 cuts
- $n \geq 5$ 
  - only procedures requiring an unbounded number of cuts are known [Brams&Taylor 1995]

# Recap

- Indivisible goods
  - house allocation: serial dictatorship
  - housing market: Top trading cycle (TTC)
- Divisible goods (cake cutting)
  - $n = 2$ : cut-and-choose
  - discrete and continuous procedures that satisfies proportionality
  - hard to design a procedure that satisfies envy-freeness

# Next class: Judgment aggregation

|                 | Action P | Action Q | Liabe? ( $P \wedge Q$ ) |
|-----------------|----------|----------|-------------------------|
| Judge 1         | Y        | Y        | Y                       |
| Judge 2         | Y        | N        | N                       |
| Judge 3         | N        | Y        | N                       |
| <b>Majority</b> | <b>Y</b> | <b>Y</b> | <b>N</b>                |