(some) Mechanisms Without Money

Algorithmic Game Theory ’22

ΑΛΜΑ, ΣΗΜΜΥ

Outline

1. Voting
2. Stable Matching
3. Top Trading Cycles

Examples

- Government policy making and referenda
  - A municipality is considering implementing a public project
  - Q1: Should we build a new road, a library or a tennis court?
  - Q2: If we build a library where shall we build it?
  - Citizens can express their preferences in an online survey or a referendum
  - Social choice: the decision of the municipality on what and where to implement

Specifying preferences

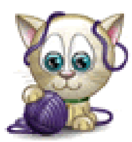
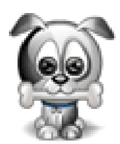
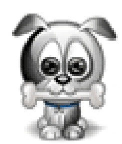
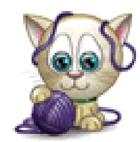
- In all the examples, the players need to submit their preferences in some form
- By ranking, by a utility function, etc
- To illustrate some of the limitations of mechanism design, we will focus first on elections and preferences by ranking

Elections setup:

- a set of candidates/alternatives $C = \{c_1, c_2, \ldots, c_m\}$
- a set of voters $V = \{1, \ldots, n\}$
- For each voter i, a preference order $>_i$
- E.g. $c_3 >_i c_1$ means that voter i prefers candidate c_3 to c_1

Elections with 2 candidates

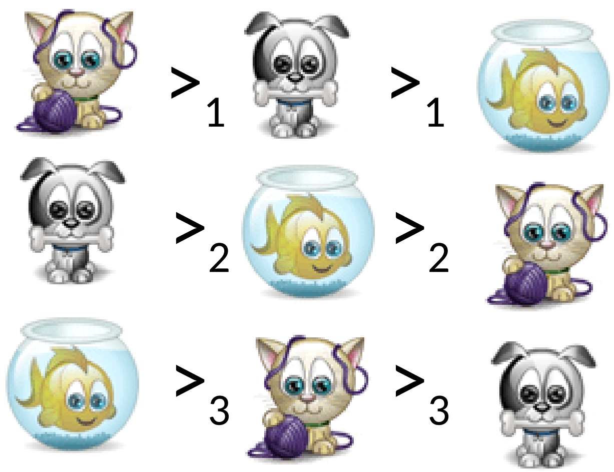
- With 2 candidates, each voter only needs to specify which one is his favorite candidate
- Suppose a family is trying to decide between getting a cat or a dog via an election
- Possible votes for voter i:

 $>_i$  or  $>_i$ 

- If we use the majority rule, no voter would have an incentive to lie about his favorite candidate
- Hence, majority voting is an appropriate social choice function when there are only 2 candidates

Elections with >=3 candidates

- Suppose now a 3rd choice of getting a fish is added
- Suppose we had a family with 1 kid, with the preferences:



Condorcet's paradox [Marquis de Condorcet 1785]:

•No matter which choice we make, a majority of voters prefer a different outcome

Lesson learnt: with >= 3 alternatives, we need to think more about how to design voting rules and what properties we want to satisfy

Social choice theory

Mechanisms for social choice problems

A mechanism corresponds to designing a function that aggregates individual preferences

Setting:

C : Set of alternatives ("candidates")

L : the set of total orders on C *(all permutations)*

Social Welfare Function: $f : L^n \rightarrow L$
It aggregates individual rankings into a global ranking
-E.g., Eurovision

Social Choice Function: $f : L^n \rightarrow C$
It aggregates individual rankings into a single winner
-E.g., elections for chair of a committee, for mayor, etc

2 impossibility results in social choice theory

1. Arrow's theorem for social welfare functions
2. Gibbard-Satterthwaite theorem for social choice functions

Social welfare functions

Some natural properties we could demand

- Unanimity: When all voters vote the same ranking, the output should be the common vote
- Independence of irrelevant alternatives (IIA): if in 2 different voting profiles, a is ranked lower than b by all voters, then the output should not depend on how other alternatives are ranked. E.g., if one voter changes his ranking for another alternative, this should not change the relative ranking between a and b in the final outcome

What type of mechanisms satisfy these axioms?

Arrow’s impossibility theorem

Definition: A social welfare function is a dictatorship if there is a voter i, such that for every voting profile, the output is identical to the preferences of voter i

Voter i is then called a dictator

Theorem [Arrow ’51]:

Every social welfare function on a set C of at least 3 alternatives that satisfies unanimity and IIA is a dictatorship.

Arrow’s theorem can be used to prove a strong negative result about strategic manipulation of elections

Social choice functions

Let's move to single-winner elections

Definition: A social choice function f is incentive compatible (or strategyproof or truthful) if for every voter i, it is a dominant strategy to submit his real ranking

Formally, for every voter i, and every profile $(>_i, >_{-i})$, it should hold that

$$f(>_i, >_{-i}) >_i f(>'_i, >_{-i}) \text{ for any dishonest ranking } >'_i$$

If f is not incentive compatible, we will say it is manipulable

- In this case, some voter would have an incentive to lie about her preferences

The Gibbard-Satterthwaite theorem

[Gibbard '73, Satterthwaite '75]
If f is an incentive compatible and onto social choice function for a set of alternatives C, with $|C| \geq 3$, then f is a dictatorship.

Very strong impossibility for voting rules

- The fundamental goal of mechanism design to avoid strategic voting behavior is a utopia (for elections where voters express preferences by a ranking)
- Incentive compatibility is too much to ask for!

A simple example

The Plurality voting rule

-Given the rankings of the voters, look only at the top choice of each voter's ranking

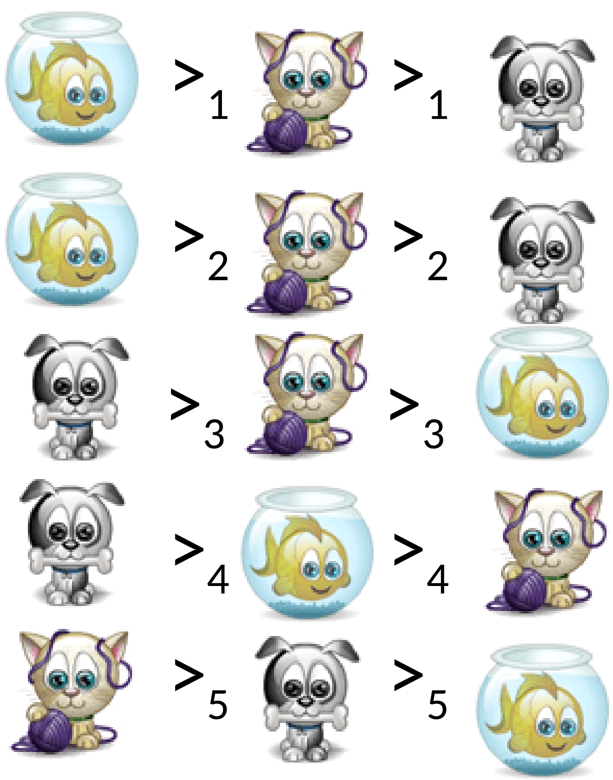
-Count for each candidate how many times they appear as a top choice

-The candidate with the highest number wins

It is very easy to construct instances where the Plurality rule can be manipulated

A simple example

Consider a family with 3 kids and the following preferences for buying a pet



- The parents prefer to get a fish (less trouble for them)
- 2 kids prefer a dog
- The 3rd kid can manipulate the election
  - Her first choice is a cat, which is not going to win
  - She also prefers the dog to the fish
  - So, she can lie and vote the dog as a first choice
  - This way the final outcome is more preferable to her

Other real life examples

- In the US presidential elections, there are actually more parties running than just Democrats and Republicans
  - The green party, Libertarians, etc
  - However, many green party supporters end up voting for Democrats to avoid a victory of the Republican candidate
  - Especially after the elections in 2000
- In general, it is very common that voters end up selecting their 2nd most preferred candidate when they realize that their top choice does not have a chance

Matchings

Match (optimally) a set of applicants to a set of open positions.

- Applicants to summer internships
- Applicants to graduate school
- Medical school graduate applicants to residency programs
- Eligible males wanting to marry eligible females

Input: males and females with their preference lists

- Every male has a preference list for women
- Every female has a preference list for men

Output: a matching with specific properties

Stablity and Instability

Consider a matching S between men and women

Unstable Pair

Male x and female y are unstable in S if:

- x prefers y to its matched female
- y prefers x to its matched male

Stable Matching

S is stable if there are no unstable pairs in S.

Formulating the Problem

Consider a set $M = \{m_1, \ldots, m_n\}$ of n men and a set $W = \{w_1, \ldots, w_n\}$ of n women.

- A matching S is a set of ordered pairs, each from $M \times W$, s.t. each member of M and each member of W appears in at most one pair in S.
- A perfect matching S' is a matching s.t. each member of M and each member of W appears in exactly one pair in S'.
- Each man $m \in M$ ranks all of the women; m prefers w to w' if m ranks w higher than w'. We refer to the ordered ranking of m as his preference list.
- Each woman ranks all of the men in the same way.
- An instability results when a perfect matching S contains two pairs (m, w) and (m', w') s.t. m prefers w' to w and w' prefers m to m'.

GOAL: A perfect matching with no instabilities.

An Example

Is the assignment X-C, Y-B, Z-A stable?

	1st (favorite)	2nd	3rd (least favorite)
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	1st (favorite)	2nd	3rd (least favorite)
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

No. Bertha and Xavier would hook up.

Questions About Stable Marriage

1. Does there exist a stable matching for every set of preference lists?
2. Given a set of preference lists, can we efficiently construct a stable matching if there is one?

The Gale-Shapley Algorithm

Initially set all $m \in M$ and $w \in W$ to free.
While $\exists m$ who is free and hasn't proposed to every $w \in W$ do
- Choose such a man m;
- w is highest ranked in m's preference list to whom m has not yet proposed
- If w is free
  then (m, w) become engaged
  else let m' be his current match
- If w prefers m' to m
  then m remains free
  else (m, w) become engaged and m' becomes free

endWhile
return the set S of engaged pairs

But Does it Work?

Some Axioms

- w remains engaged from the point at which she receives her first proposal
- the sequence of partners with which w is engaged gets increasingly better (in terms of her preference list)
- the sequence of women to whom m proposes get increasingly worse (in terms of his preference list)

Men propose to women in decreasing order of preference (men "optimistic").

Once a woman is matched, she never becomes unmatched (only "trades up").

Termination

Theorem

The G-S algorithm terminates after at most n^2 iterations of the while loop.

What is a good measure of progress?

- the number of free men?
- the number of engaged couples?
- the number of proposals made?

Proof by counting proposals

- Each iteration consists of one man proposing to a woman he has never proposed to before.
- After each iteration of the while loop, the number of proposals increases by one
- Every man proposes at most once to a woman: $|proposals| \leq n^2$

A Perfect Matching Returned

Theorem

The set S returned at termination is a perfect matching.

Proof

- It is a matching since it only trades pairs with the same woman
- Women only trade up, thus once matched, remain matched.
- There is no free man at the end: He has proposed to all women so all of them should be matched.

and Stable

Theorem

If the algorithm return a matching S, then S is a stable matching.

Proof (by contradiction)

- Let pairs (m, w) and (m', w') in S be s.t.
  - m prefers w' to w, i.e., $w' >_m w$, and
  - w' prefers m to m', i.e., $m >_{w'} m'$.
- m proposed to w' in the past and at some point got rejected for m''.
- In the preference list of w': $m'' >_{w'} m$ and $m' \geq_{w'} m''$.
- m is below m' in the preference list of w', contradiction.

Summary

The Gale-Shapley algorithm guarantees to find a stable matching.

- Are there multiple stable matchings?
- If multiple stable matchings, which to choose??
- Which one does the algorithm find? (Any properties?)

Understanding the Solution

For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?

An instance with two stable matchings:
A-X, B-Y, C-Z
A-Y, B-X, C-Z

	1st	2nd	3rd
Xavier	A	B	C
Yancey	B	A	C
Zeus	A	B	C

	1st	2nd	3rd
Amy	Y	X	Z
Bertha	X	Y	Z
Clare	X	Y	Z

Proposer Optimal Solution Returned

- Man m and woman w are valid partners if there exists some stable matching in which they are matched
- A man-optimal matching is one in which every man receives the best valid partner

- Claim 1: All executions of GS yield man-optimal assignment, which is a stable matching.
- Claim 2: All executions of GS yield woman-pessimal assignment, which is a stable matching (i.e., each woman receives the worst possible valid partner).

Claim 1: man-optimality

By contradiction: Let S' be a stable matching where m is better off.

- Let (m, w) be a pair in S'
- In the algorithm m proposed to w and got rejected for some m', thus

$$m' >_w m$$

- Assume this is the first rejection by a valid partner
- Let (m', w') be a pair in $S' \Rightarrow w'$ is valid for m'.
- In the algorithm m' did not get rejected by a valid partner before m did $\Rightarrow m'$ proposed to w before any valid woman, thus

$$w >_{m'} w'$$

- S' not stable: $[(m, w) \in S']$ & $[(m', w') \in S']$ & $[m' >_w m]$ & $[w >_{m'} w']$

Claim 2: woman-pessimality

By contradiction: Let S be the algorithm's matching

- Let $(m, w) \in S$ and m not worst valid for w.
- Exists S' with $(m', w) \in S'$ and

$$m >_w m'$$

- Let $(m, w') \in S'$ be partner of m in S'. By man optimality

$$w >_m w'$$

- S' not stable: $[(m, w) \in S']$ & $[(m', w') \in S']$ & $[m' >_w m]$ & $[w >_{m'} w']$

Incentives - Strategy Proofness

Slight extension where players can mark others as unacceptable

- Truthtelling is still proposer-optimal
- Proposal-receivers may benefit by misreporting

Truthful reporting

Albert	Diane	Emily
Bradley	Emily	Diane

Diane	Bradley	Albert
Emily	Albert	Bradley

Albert	Diane	Emily
Bradley	Emily	Diane

Diane	Bradley	Albert
Emily	Albert	Bradley

Strategic reporting

Albert	Diane	Emily
Bradley	Emily	Diane

Diane	Bradley	⊘
Emily	Albert	Bradley

Albert	Diane	Emily
Bradley	Emily	Diane

Diane	Bradley	⊘
Emily	Albert	Bradley

Impossibility results

There is no matching mechanism that

1. is strategy proof for both sides and
2. always results in a stable outcome (given revealed preferences)

Consider a many-to-one extension where "men" can have up to q "women" (classes and students)

These problems look very similar yet

- No algorithm exists s.t. truthtelling is dominant strategy for "men"

Leaving Bipartite Graphs

Consider the stable roommate problem. $2n$ people each rank the others from 1 to $2n-1$. The goal is to assign roommate pairs so that none are unstable.

	1st	2nd	3rd
Adam	B	C	D
Bob	C	A	D
Chris	A	B	D
Doofus	A	B	C

A-B, C-D $\Rightarrow$ B-C unstable
A-C, B-D $\Rightarrow$ A-B unstable
A-D, B-C $\Rightarrow$ A-C unstable

Observation: a stable matching doesn't always exist.

Irving 1985

There exists an algorithm returning a matching or deciding non existence.
(Builds on Gale-Shapley ideas and work by McVitie and Wilson '71)

Trading Houses

The problem

- n players own n houses.
- Each player has strict preferences over houses.
- Can the players benefit from swapping houses?

The Top Trading Cycles algorithm

1. Each player points to her most preferred house (maybe its own).
2. Each house points back to its owner.
3. In the directed graph identify cycles.
   - outdegree 1 & finite number of players $\rightarrow$ cycles exist
   - outdegree 1 $\rightarrow$ each player to at most one cycle
4. Give each player in a cycle the house she points at and remove her and her assigned house.
5. Repeat until there are no unmatched players/houses.

Some Nice Properties

Claim 1

No coallition can make all of its members better off by exchanging the houses: TTC returns a core allocation

Claim 2

Given initial houses allocation, there is only one such assignment that the players accept: unique core allocation

Claim 3

When TTC is used, no advantage for a player to lie: Strategy-proofness.

Claim 1: Core Allocation

Proof of Claim 1

Let N_j denote the players allocated in the j-th iteration of the algorithm.

- Assume players report truthfully and let S be a coalition.
- Let ℓ be the first iteration for which some $i \in S$ got a house: $i \in S \cap N_\ell$.
- No player of S belongs to $N_1, \ldots, N_{\ell-1} \Rightarrow$ no S-reallocation "improves" i.

Claim 2: Uniqueness of Core Allocation

Proof of Claim 2

Let N_j denote the players allocated in the j-th iteration of the algorithm.

- Let H_j be the houses remaining after the $(j-1)$-th iteration
- Consider any other core allocation A
- Let ℓ be the smallest index j for which some $i \in N_j$ does not receive her first choice among H_j in A.
- Let C be a cycle for players in N_ℓ containing i
- Players in C may change according to C and "improve", a contradiction.

Claim 3: Dominant Strategy Incentives Compatible (DSIC)

Proof of Claim 3

Let N_j denote the players allocated in the j-th iteration, under truthfullness.

- Let ℓ be the smallest index for which an player in N_ℓ has incentive to misreport
- The algorithm will assign the same houses to all players in $\cup_{j<\ell} N_j$
- Let H_j denote the houses remaining after the $(j-1)$-th iteration
- $i \in N_\ell$ by misreporting will not take a house in $H_1 \setminus H_\ell$
- $i \in N_\ell$ will take her most preferred house in H_ℓ, a contradiction

Many-to-one Extension

Assignments of students to schools.

- Students submit strict preferences over schools.
- Schools submit strict preferences over students based on priority criteria (and a random number generator)

Modified TTC algorithm

1. Each student points at her most preferred unfilled school
2. Each school points at its most preferred student.
3. Cycles are identified and students in cycles are matched to the school they point at.
4. Remove assigned students and full schools.
5. Repeat if there are unassigned students