CS639: Algorithmic Game Theory & Learning

Chapter 6: Mechanism Design without Money

Kirthevasan Kandasamy

UW-Madison

Introduction to mechanism design

Game theory. So far in class: the game (an interaction between agents) is given:

- Input: the game (each agent's actions, utilities etc).
- Goal: predict behavior of self-interested, rational agents (players).
- Process: apply solution concepts such as DSE, NE, CE, and CCE.

Mechanism design. In the next two chapters, we will study *mechanism design*, which can be viewed as "inverse game theory".

- Input: a desired social goal (*e.g.*, find good matchings of residents to hospitals or students to schools, match organ donors to patients, maximize welfare in social planning projects, maximize revenue for a seller).
- Goal: design a game that, when played by rational agents, achieves this goal.
- Process: design the rules of interaction to incentivize "good" behavior.

Overarching challenges in mechanism design

1. **Incentive compatibility:** An agent's utility depends on some *private* information that the mechanism designer does not observe.
   - The designer can ask agents to report it, but strategic agents may lie to obtain higher utility. An agent's actions are the space of available reports.
   - A mechanism is **incentive compatible** if every agent finds it optimal to report their true information.
2. **Individual rationality:** Ensure that agents are better off participating in the mechanism than not participating.
3. **Efficiency/Stability:** Ensure that when self-interested agents play the game, the desired social goal is achieved.

Revisit this slide after each mechanism to understand the challenges in each setting and how we navigate them.

Overview of Chapters 6 and 7

Chapter 6: Mechanism design without money. Useful when monetary transfers are illegal, unethical, or impractical (*e.g.*, organ donation, public schools).

- Stable matching
- One-sided matching
- Fair resource allocation

Nobel Prize in Economics 2012, Alvin Roth & Lloyd Shapley

Chapter 7: Mechanism design with money. Used when payments to/from agents can help align incentives and efficiently allocate resources.

- Social welfare maximization
- Revenue maximization in single-parameter auctions
- Posted price mechanisms and prophet inequality

Nobel Prize in Economics 2007, Leonid Hurwicz, Eric Maskin, & Roger Myerson,
Nobel Prize in Economics 2020, Paul Milgrom & Robert Wilson

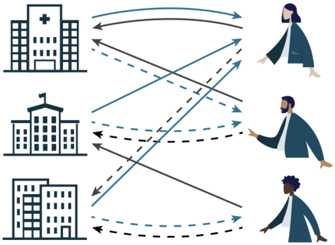
Outline

1. Stable matching
2. One-sided matching
3. Fair resource allocation

Slides are intended as teaching aids only and do not include all material discussed in class. Students are strongly encouraged to attend lectures and take their own notes.

Ch 6.1: Stable matching

Example: National residency matching program.



There are a set of residents and hospitals

- Each resident has a preference ordering over hospitals.
- Each hospital has a preference ordering over residents.

We wish to match residents to hospitals (each resident is assigned to at most one hospital, and each hospital is not assigned more than it can accommdate).

Stable matching (cont'd)

What do we want from such a matching algorithm?

- The matching is "stable": no resident-hospital pair would rather be matched with each other than their current assignment.
- No resident should have an incentive to misrepresent her preferences (e.g., by reporting that she prefers hospital A over Hospital B when, in fact, she prefers the opposite).

Real-world examples of stable matching

- Residency matching
- School/college admissions
- Teacher transfers in public school systems

Stable matching (cont'd)

Model. There are two sides (sets of agents) in a market. We wish to match agents on side to agents in another.

- Let $\mathcal{F}$ denote a set of firms and $\mathcal{W}$ denote a set of workers.

- Say $|\mathcal{F}| = |\mathcal{W}| = n$, and that each firm has exactly one slot for a worker.

- **Matching:** A matching $\mu : \mathcal{F} \to \mathcal{W}$ is a bijection from firms to workers. Let $\mu^{-1} : \mathcal{W} \to \mathcal{F}$ be the inverse of μ.

- **Preferences:** Each firm f has a strict total ordering over $\mathcal{W}$. If f prefers worker w to w', we will write $w \succ_f w'$. We will denote firm f's preferences p_f.
Similarly, each worker w has a strict total ordering of preferences over $\mathcal{F}$, denoted p_w.

[0] With some work, we can extend this framework to handle $|\mathcal{F}| \neq |\mathcal{W}|$ and non-strict preferences.

Stable matching (cont'd)

Definition (Stability). In a matching μ, a firm-worker pair (f, w) who are not matched to each other (*i.e.*, $\mu(f) \neq w$) are called a *blocking pair* if f and w prefer each other to their matches in μ, *i.e.*, $w \succ_f \mu(f)$ and $f \succ_w \mu^{-1}(w)$.
A matching μ is *stable* if it has no blocking pairs.

Example 1. Let $\mathcal{F} = \{a, b\}$ and $\mathcal{W} = \{x, y\}$. Suppose

$$p_a : x \succ_a y, \qquad p_x : b \succ_x a,$$
$$p_b : x \succ_b y, \qquad p_y : a \succ_y b.$$

Here, $a \leftrightarrow x$, $b \leftrightarrow y$ is unstable since (b, x) is a blocking pair.
But $a \leftrightarrow y$, $b \leftrightarrow x$ is stable.

Stable matching examples (cont'd)

Example 2. Let $\mathcal{F} = \{a, b\}$ and $\mathcal{W} = \{x, y\}$. Suppose

$$p_a : x \succ_a y, \qquad p_x : b \succ_x a,$$
$$p_b : y \succ_b x, \qquad p_y : a \succ_y b.$$

Here, $a \leftrightarrow x$, $b \leftrightarrow y$ is stable.
Similarly, $a \leftrightarrow y$, $b \leftrightarrow x$ is also stable.

Example 3 (Figure 10.2 in KP). Let $\mathcal{F} = \{a, b, c\}$ and $\mathcal{W} = \{x, y, z\}$. Suppose

$$p_a : y \succ_a z \succ_a x, \qquad p_x : a \succ_x b \succ_x c,$$
$$p_b : y \succ_b z \succ_b x, \qquad p_y : b \succ_y c \succ_y a,$$
$$p_c : x \succ_c y \succ_c z, \qquad p_z : a \succ_z c \succ_z b.$$

Here, $a \leftrightarrow x$, $b \leftrightarrow y$, $c \leftrightarrow z$ is unstable since (a, z) is a blocking pair.
But $b \leftrightarrow y$, $c \leftrightarrow x$, $a \leftrightarrow z$ is stable.

The Gale-Shapley (a.k.a Deferred acceptance) algorithm

Firm-proposing version of Gale-Shapley. In a market in which workers have applied to firms, and firms are now attempting to recruit their most preferred applicants.

- Given: firm preferences $\{p_f\}_{f \in \mathcal{F}}$, worker preferences $\{p_w\}_{w \in \mathcal{W}}$.
- Initially all firms and workers are unmatched.
- **While** there exists an unmatched firm f:
  - f proposes to its most preferred worker w that has not rejected it yet.
  - **If** w is currently unmatched:
    w accepts the proposal (w and f are temporarily matched).
  - **Else if** w is currently matched to f' and prefers f to f', *i.e.*, $f \succ_w f'$:
    w accepts the proposal from f and rejects f' (w and f are temporarily matched, f' becomes unmatched).
  - **Else if** w is currently matched to f' and prefers f' to f, *i.e.*, $f' \succ_w f$:
    w rejects the proposal from f (f continues to remain unmatched).
- **Return** the current matching.

We can similarly define a worker proposing version of GS.

The Gale-Shapley algorithm (cont'd)

Theorem (Theorem 10.2.3 in KP)). On any set of preferences $\{p_f\}_{f \in \mathcal{F}}$, $\{p_w\}_{w \in \mathcal{W}}$, Gale-Shapley terminates and returns a stable matching.

Proof. On each round there is exactly one proposal. But there can be at most n^2 proposals since a firm never proposes to a worker who has rejected it. Hence, GS terminates in at most n^2 rounds.

- Let μ denote the matching returned by GS.

- By way of contradiction, suppose that (f, w) is a blocking pair for μ, *i.e.*, they prefer each other over their matches in μ. That is,

$$w \succ_f \mu(f), \qquad f \succ_w \mu^{-1}(w). \tag{1}$$

The Gale-Shapley algorithm (cont'd)

- Since $w \succ_f \mu(f)$, f must have proposed to w before proposing to $\mu(f)$. But w rejected f in favor of some f'. Then, either

$$f' = \mu^{-1}(w) \succ_w f \qquad \text{or} \qquad \mu^{-1}(w) \succ_w f' \succ_w f.$$

- Both of these contradict $f \succ_w \mu^{-1}(w)$ in (1). $\square$

Attainable pairs in Gale-Shapley

The following is an important property of Gale-Shapley. To state this, let us first define attainability.

Definition (Attainability). We say that a firm f is attainable for a worker w if there exists a stable matching $\widetilde{\mu}$ such that $\widetilde{\mu}(f) = w$. Similarly, a worker w is attainable for a firm f if there exists a stable matching $\widetilde{\mu}$ such that $\widetilde{\mu}^{-1}(w) = f$.

Theorem (Theorem 10.3.1 in KP). Let μ be the stable matching produced by the firm-proposing version of Gale-Shapley. Then:

(i) Every firm is matched to its most preferred attainable worker.

(ii) Every worker is matched to its least preferred attainable firm.

Attainable pairs in Gale-Shapley (cont'd)

Proof. We will prove *(i)* in class. You will prove *(ii)* in the homework.

By way of contradiction, assume that there is at least one firm which is not matched to its most preferred attainable worker. Since each firm's proposals are in order of its preferences, it is sufficient to show that no firm is rejected by an attainable worker.

- Consider the *first round* in which *any firm* is rejected by an attainable worker in Gale-Shapley. Call this firm f and let w be the (most preferred) attainable worker which rejected f.

- Let f' be the firm that w rejected f in favor of. Hence $f' \succ_w f$.

- Since w is attainable for f, there exists some stable matching $\widetilde{\mu}$ in which $\widetilde{\mu}(f) = w$. Let $\widetilde{\mu}(f') = w'$ for some $w' \neq w$.

Attainable pairs in Gale-Shapley (cont'd)

Recall, we have shown: *(a)* $f' \succ_w f$, and
(b) there exists a stable matching $\widetilde{\mu}$ with $\widetilde{\mu}(f) = w$, and $\widetilde{\mu}(f') = w'$ for some worker w'.

Claim. We have, $w \succ_{f'} w'$.

Proof of claim: By way of contradiction, suppose $w' \succ_{f'} w$. Then f' will have proposed to w' first in Gale-Shapley, and must have been rejected by w' so that it can propose to w. But we are assuming that this is the first time any firm is rejected by an attainable worker. □

(Continuing with our main proof) But this means that $\widetilde{\mu}$ is not a stable matching. In particular, recall that we have $\widetilde{\mu}(f) = w$, and $\widetilde{\mu}(f') = w'$, yet we have shown

$$f' \succ_w f = \widetilde{\mu}^{-1}(w), \qquad w \succ_{f'} w' = \widetilde{\mu}(f')$$

Therefore, (f', w) are a blocking pair for $\widetilde{\mu}$. □

Attainable pairs in Gale-Shapley (cont'd)

Recall, Theorem: Let μ be the stable matching produced by the firm-proposing version of Gale-Shapley. Then:

(i) Every firm is matched to its most preferred attainable worker.

(ii) Every worker is matched to its least preferred attainable firm.

Corollary. Suppose (f, w) get matched in the firm-proposing and worker proposing versions GS. Then, w is the only attainable worker for f, and f is the only attainable firm for w.

Proof. w is the most and least preferred attainable worker for f. □

Mechanism design for stable matching

So far, we have assumed that agents' preferences are given.

In reality, we have to rely on firms and workers to *report* their preferences, and that they are *selfish, strategic agents* who may misreport in order to obtain a better match.

Each agent's "action" is a bid (b_f or b_w) that she submits representing her true preferences (p_f or p_w). These bids need not be truthful.

An algorithm that *elicits preferences* and *computes a matching* is called a *mechanism* for stable matching. What do we want from a mechanism?

(i) Incentive-compatibility (IC): the best strategy for each agent should be to submit their preferences truthfully (*e.g.*, $b_f = p_f$ for a firm).

(ii) Stability: outputs a stable matching provided that all agents have submitted truthfully.

On incentive compatibility

In mechanism design (generally, not just in stable matching), we may formulate IC in different ways. The most common, and strongest, is DSIC.

Dominant strategy incentive compatibility (DSIC). The best strategy for an agent is to report truthfully regardless of how others are reporting. That is, truthful reporting is a dominant strategy for each agent.

In this class, we will primarily focus on DSIC. But DSIC is not always possible so one may consider weaker notions, such as the one below (other formulations are also possible).

Nash incentive compatibility. Truthful reporting is a NE in the mechanism, *i.e.*, best strategy is to report truthfully when others are also reporting truthfully.

Terminology. When we say a mechanism is DSIC/strategy-proof/truthful, they mean the same thing.

Mechanism design for stable matching (cont'd)

Consider the following mechanism:
- Ask agents to report their bids (preferences).
- Execute the firm-proposing version of Gale-Shapley on the bids.

Theorem. The firm-proposing version of Gale-Shapley is:

(i) Stable when all agents report their true preferences.

(ii) DSIC for firms. That is, regardless of the bids of the other agents (firms or workers), it is best for each firm f to report $b_f = p_f$.

(iii) Not DSIC for workers. That is, there exists an instance $(\mathcal{F}, \mathcal{W}, \{p_f\}_{f \in \mathcal{F}}, \{p_w\}_{w \in \mathcal{W}})$ where a worker $w \in \mathcal{W}$ can obtain a better match by reporting $b_w \neq p_w$ when others are truthful.

Limits of IC in stable matching

That Gale-Shapley is not DSIC for workers may seem concerning at first. But this is tempered by the following observations:

1) It is impossible to elicit truthful preferences from both sides of the market simultaneously.

> **Theorem (Roth, 1982).** There is no mechanism which always returns a stable matching and is DSIC for both firms and workers.

2) In practice, we may only need DSIC on one side of the market. For instance, when matching residents to hospitals, we want to prevent strategic behavior from residents, but can reasonably trust hospitals to report honestly. A similar reasoning applies when matching students to schools.

Proof of theorem

Recall, Theorem. The firm-proposing version of Gale-Shapley is:
(i) Stable when all agents report their true preferences.
(ii) DSIC for firms. That is, regardless of the bids of the other agents (firms or workers), it is best for each firm f to report $b_f = p_f$.
(iii) Not DSIC for workers. That is, there exists an instance $(\mathcal{F}, \mathcal{W}, \{p_f\}_{f\in\mathcal{F}}, \{p_w\}_{w\in\mathcal{W}})$ where a worker $w \in \mathcal{W}$ can obtain a better match by reporting $b_w \neq p_w$ when others are truthful.

Proof. We have already proved *(i)*. You will prove *(iii)* in the homework.

Proof of (ii). Consider a firm f. Let us fix the reported preferences (bids) of *others* $(\{p_g\}_{g\in\mathcal{F}\setminus\{f\}}, \{p_w\}_{w\in\mathcal{W}})$. Without loss of generality, we can simply pretend that these are their true preferences[1].

Let p_f be the true preferences of f, and let p'_f be any other (nontruthful) bid.

[1]With the understanding that when we write $a \succ_b c$ for any $b \in \{\mathcal{F}, \mathcal{W}\} \setminus \{f\}$, it will refer to preferences under their reported bids.

Proof of IC (cont'd)

Denote $p = (p_f, p_{-f})$ and $p' = (p'_f, p_{-f})$. Let μ be the matching returned by Gale-Shapley under p and ν be the matching under p'.

We need to show that reporting $b_f = p_f$ yields a no worse match than reporting $b_f = p'_f$. That is, $\mu(f) = \nu(f)$ or $\mu(f) \succ_f \nu(f)$ (recall that we are assuming strict preferences). By way of contradiction, suppose $\nu(f) \succ_f \mu(f)$.

Let $S \triangleq \{f' \in \mathcal{F}; \nu(f') \succ_{f'} \mu(f')\}$ be the firms who prefer their matches in ν to μ. Clearly, $S \neq \varnothing$ since $f \in S$.

We will first establish the following claim about the projections

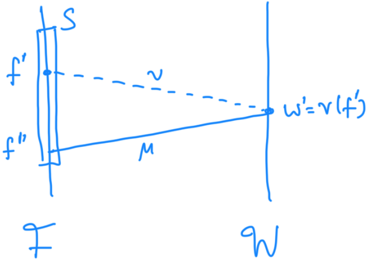
$$\nu(S) = \{w; \nu^{-1}(w) \in S\}, \qquad \mu(S) = \{w; \mu^{-1}(w) \in S\}.$$

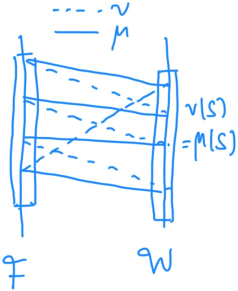
Claim 1. $\mu(S) = \nu(S)$.

Proof of Claim 1

Claim 1. Let $\nu(S) = \{w; \nu^{-1}(w) \in S\}$, and $\mu(S) = \{w; \mu^{-1}(w) \in S\}$. Then, $\mu(S) = \nu(S)$.

Proof of Claim 1. We will prove the following equivalent statement: Let $f' \in S$, $w' = \nu(f')$, and $f'' = \mu^{-1}(w')$. Then $f'' \in S$.





If $f'' = f$, then $f \in S$. Therefore, say $f'' \neq f$. This means that f'' has reported truthfully.

Since $f' \in S$, $w' = \nu(f') \succ_{f'} \mu(f')$.

Proof of Claim 1 (cont'd)

If w' also prefers $f' = \nu^{-1}(f')$ to $\mu^{-1}(w')$, then μ would be unstable as (f', w') would be a blocking pair for μ. Hence,

$$f'' = \mu^{-1}(w') \succ_{w'} \nu^{-1}(w') = f'.$$

If f'' also prefers w' to $\nu(f'')$, then ν would be unstable as (f'', w') would be a blocking pair. Hence,

$$\nu(f'') \succ_{f''} w' = \mu(f'').$$

Hence, $f'' \in S$. This implies $\nu(S) = \mu(S)$ $\square$

Proof of IC (cont'd)

Recall: *(i)* GS under $p = (p_f, p_{-f})$ produces μ, GS with $p' = (p', p_{-f})$ produces ν.
(ii) $S \triangleq \{f' \in \mathcal{F}; \nu(f') \succ_{f'} \mu(f')\}$. *(ii) Claim 1:* $\mu(S) = \nu(S)$.

We know $f \in S$. If $S = \{f\}$ is a singleton, then $\nu(S) = \mu(S)$ by Claim 1, and hence $\nu(f) = \mu(f)$. This is a contradiction to $\nu(f) \succ_f \mu(f)$. Hence, going forward, we will assume that there is at least another firm in S.

Now consider GS under $p = (p_f, p_{-f})$. By *(ii)* and the fact that each $f' \in S$ was matched to $\mu(f')$, we have that each $f' \in S$ proposed to $\nu(f')$ and was rejected by $\nu(f')$ in some round. Let f_1 be the last firm in S who proposes. This proposal is necessarily to $\mu(f_1) = w_1$ (say). We have,

$$w_1 \in \mu(S) \implies w_1 \in \nu(S) \qquad \texttt{By Claim 1}$$

$$\implies f_2 \triangleq \nu^{-1}(w_1) \in S$$

$$\implies w_1 = \nu(f_2) \succ_{f_2} \mu(f_2). \qquad \texttt{definition of } S$$

Therefore, f_2 must have proposed to w_1 before f_1 proposed (and was matched) to w_1.

Proof of IC (cont'd)

We can draw the following conclusions in GS under $p = (p_f, p_{-f})$:

- f_2 proposed and was permanently matched to $\mu(f_2)$ before f_1 proposed to $\mu(f_1) = w_1$ (as f_1 was the last in S to propose).

- When f_1 proposed to w_1, w_1 rejected some other agent f_3 such that

$$f_3 \succ_{w_1} f_2 = \nu^{-1}(w_1)$$

- But, by a similar reasoning, we have $f_3 \notin S$. If $f_3 \in S$, then f_3 will propose again and f_1 would not be the last in S to propose.

Proof of IC (cont'd)

Recall: *(i)* $S \triangleq \{f' \in \mathcal{F}; \nu(f') \succ_{f'} \mu(f')\}$. *(ii)* We have shown $f_3 \succ_{w_1} f_2 = \nu^{-1}(w_1)$.

We therefore have,

$$w_1 \overset{(a)}{\succ}_{f_3} \mu(f_3) \overset{(b)}{\succeq}_{f_3} \nu(f_3) \qquad (*)$$

Here, (a) uses the observation that f_3 proposed to w_1 before $\mu(f_3)$, and (b) follows from the fact that $f_3 \notin S$. (Here, $a \succeq_b c$ means $a \succ_b c$ or $a = c$.)

From *(ii)* and $(*)$ we have that (f_3, w_1) are a blocking pair for ν. But ν is stable under $p' = (p'_f, p_{-f})$ which is a contradiction. Note that as $f_3 \notin S$, $f_3 \neq f$ and hence p_{f_3} are true preferences, as are p_{w_1}. □

Ch 6.2: One-sided matching

Example. Consider the following example:

- There are n students, each initially assigned an apartment in university housing.
- However, two students might prefer each other's assigned apartments to their own.
- More generally, a group of students might be able to swap apartments among themselves so that each obtains a more preferred apartment.

Can we design an algorithm to reallocate the apartments?

One-sided matching (cont'd)

What properties do we want from an apartment reallocation mechanism?

- Incentive compatibility: No student should have an incentive to misreport her preferences (ranking over the apartments).
- Individual rationality: No student should be worse off by participating in the mechanism, *i.e.*, a student should not be assigned an apartment she prefers less than her initial allocation.
- Stability: no group of students should be able to simultaneously switch apartments so that everyone in the group is better off.

Real-world examples of one-sided matching:

- Kidney exchange
- Social housing reallocation
- Trading shifts in hospitals
- Student housing at many universities

One-sided matching (cont'd)

Model. There are a set of n agents, denoted $[n]$. Each agent brings a good for trade.

- Agent i has a *strict* total ordering of preferences over the n goods brought by all agents, including her own. If i prefers j's good to k's, then we write $j \succ_i k$. Denote i's preferences by p_i.

- We wish to facilitate a trade between agents.

- An allocation is a bijection $\pi : [n] \to [n]$. If $\pi(i) = j$, it means that i was allocated j's good.

In a mechanism for one-sided matching

- Each agent $i \in [n]$ reports a bid b_i representing her preferences (not necessarily truthfully).

- The mechanism allocates the goods to agents based on the reported preferences.

Stability/Pareto-efficiency in one-sided matching

Intuitively, we say that an allocation is stable if no subset of agents can exchange goods allocated to them such that all of them obtain a more preferred good. This can be formalized via the notion of Pareto-efficiency (a.k.a Pareto-optimality).

Definition (Pareto-efficiency in one-sided matching).

- Let π, π' be two allocations such that $\pi \neq \pi'$. We say π' Pareto-dominates π if for all $i \in \{j \in [n]; \pi'(j) \neq \pi(j)\}$ we have $\pi'(i) \succ_i \pi(i)$.

- We say that an allocation π is Pareto-efficient if there is no other allocation π' that Pareto-dominates π.

Desiderata for a mechanism

A mechansim π should satisfy the following three desiderata:

1. **Dominant strategy incentive compatibility (DSIC):** Regardless of the bids submitted by the others, the best strategy for each agent is to report her preferences truthfully.

2. **Individual rationality (IR):** A truthful agent should not be worse off by participating in the mechanism. That is, for any agent i reporting $b_i = p_i$, we have $\pi(i) = i$ or $\pi(i) \succ_i i$.

3. **Pareto-efficiency (PE):** When all agents report truthfully, the allocation should be Pareto-efficient.

The Top Trading Cycles (TTC) Algorithm

- Given: (reported) preferences $\{b_i\}_{i\in[n]}$.
- $t \leftarrow 1$. `# Round index.`
- Let $S_1 \leftarrow [n]$. `# S_t are agents remaining on round t.`
- $\pi(i) \leftarrow \texttt{NULL}$ for all $i \in [n]$. `# Allocations.`
- While $|S_t| > 0$:
  - Let $G_t = (S_t, E_t)$ denote the directed graph where the vertices are the surviving agents and E_t contains directed edges from each $i \in S_t$ to the agent with i's most preferred good in S_t, according to the reported preferences $\{b_i\}_{i\in[n]}$.
  - Let C_t be all edges which are part of a cycle in G_t. For each $(i,j) \in C_t$, set $\pi(i) \leftarrow j$.
  - Clear all cycles: $S_{t+1} \leftarrow S_t \setminus \{i; (i,j) \in C_t\}$.
  - $t \leftarrow t + 1$.
- Return π.

The Top Trading Cycles (TTC) Algorithm

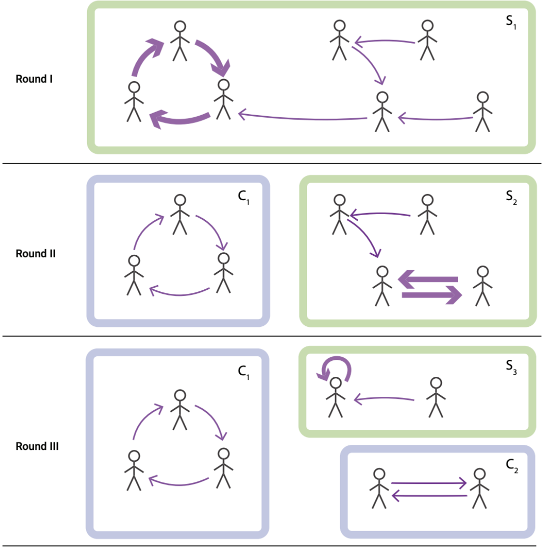


Figure 10.9 in KP.

The Top Trading Cycles (TTC) Algorithm

Theorem. TTC always terminates and satisfies DSIC, IR, and PE.

Proof. In the homework.

Kidney exchange

- Many kidney patients have a willing donor (often a relative), but the donor may be *medically incompatible* with the patient (usually due to mismatch in blood type).
- We can treat patients as "agents" and donors as "goods" and execute TTC.

Practical considerations.

- Often we have many patients without a donor and many donors with no patients (deceased donors and altruistic living donors).
- When clearing a cycle, operations need to be performed simultaneously. Otherwise, one patient-donor pair can renege on the donation after receiving a kidney. In practice, this means that cycles need to be kept short, as a cycle of length k requires $2k$ surgeries to remove the kidney from a donor and and implant to the patient. (In practice, cycles of length at most 3).

Kidney exchange (cont'd)

Practical considerations (cont'd).

- A patient's preference ordering is primarily determined by compatibility. After compatibility, it may be based on geographic proximity, health condition etc. However, not all of these factors can be modeled via TTC (*e.g.*, donor and patient may be in two different locations).
- Local transplant centers often match their easy-to-match patient-donor pairs internally and only send the hard-to-match patients (e.g., Blood Type O) to the national pool. This makes it harder to find matches for everyone.

Ch 6.3: Fair resource allocation

Go through slides 8–14 from the overview slides
for real-world motivation and an intuitive description of the challenges.

Model for fair resource allocation

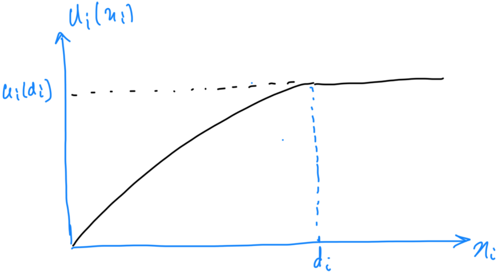
Model. There are n agents, each sharing a resource of size 1.

- Agent i's endowment is e_i, with $\sum_{i=1}^{n} e_i = 1$. We can view the endowment as agent i's contribution to the pool of resources.

An agent's demand and utility.

- If an agent is allocated x_i resources, her utility is $\widetilde{u}_i(x_i)$ where $\widetilde{u}_i : \mathbb{R}_+ \to \mathbb{R}_+$ is non-decreasing (more resources cannot hurt).
- Agent i's demand d_i is the amount of resources after which her utility does not increase. We will assume

$$\widetilde{u}_i(z) < \widetilde{u}_i(d_i) \text{ for all } z < d_i,$$
$$\widetilde{u}_i(z) = \widetilde{u}_i(d_i) \text{ for all } z \geq d_i,$$



Mechanism design for resource allocation

In a mechanism M for resource allocation:

- Each agent i will submit a bid $b_i \in \mathbb{R}_+$ representing her demand d_i (not necessarily truthfully).

- The mechanism designer will compute an allocation $x = \{x_i\}_{i\in[n]}$, where agent i receives x_i resources. The allocation must be feasible, *i.e.*, $\sum_{i=1}^{n} x_i \leq 1$.

We can write the allocation x as a function of the bids $b = \{b_i\}_{i\in[n]}$ and the mechanism M, *i.e.*,

$$x(b; M) = (x_1(b; M), \ldots, x_n(b; M)),$$

An agent's "action" is the bid she submits (action space is $\mathbb{R}_+$).

Agent's utility in a mechanism. We can write the agent's utility in a mechanism M under the bid (action) profile b as,

$$u_i(b; M) \triangleq \widetilde{u}_i(x_i(b; M)).$$

Desiderata for a mechanism

1) Dominant-strategy incentive-compatibility (DSIC). No agent should have an incentive to misreport her demand regardless of others' bids. Formally, reporting the true demand, *i.e.*, $b_i = d_i$ is a dominant strategy, *i.e.*,

$$u_i(d_i, b_{-i}; M) \geq u_i(b_i, b_{-i}; M) \qquad \text{for all } b_i \in \mathbb{R}_+,\ b_{-i} \in \mathbb{R}_+^{n-1}.$$

2) Individual rationality (IR). A truthful agent's utility is at least as much as when using her endowment, *i.e.*, when not participating in the mechanism. Formally,

$$u_i(d_i, b_{-i}; M) \geq \widetilde{u}_i(e_i), \qquad \text{for all } b_{-i} \in \mathbb{R}_+^{n-1}.$$

Desiderata for a mechanism (cont'd)

3) Efficiency. There should be no idle resources when there is an unment demand.

Let $y^+ = \max(y, 0)$. Given an allocation x, let us define the unallocated resources $\ell^{\mathrm{ur}}(x)$, over-allocated resources $\ell^{\mathrm{or}}(x)$ and unment demand $\ell^{\mathrm{ud}}(x)$ as follows:

$$\ell^{\mathrm{ur}}(x) = 1 - \sum_{i=1}^{n} x_i, \qquad \ell^{\mathrm{or}}(x) = \sum_{i=1}^{n} (x_i - d_i)^+, \qquad \ell^{\mathrm{ud}}(x) = \sum_{i=1}^{n} (d_i - x_i)^+.$$

The resource loss is the amount of resources not being used, but that could have been used to improve the utility of some agent:

$$\ell(x) = \min\left(\ell^{\mathrm{ud}}(x), \underbrace{\ell^{\mathrm{or}}(x) + \ell^{\mathrm{ur}}(x)}_{\substack{=\text{amount of resources}\\ \text{not being used}}} \right)$$

A mechanism is efficient if it achieves zero resource loss when all agents are reporting truthfully, *i.e.*, $\ell(x(d; M)) = 0$.

Zero resource loss and Pareto-efficiency

Achieving zero resource loss, while intuitive in its own right, is also equivalent to achieving Pareto-efficient (PE) agent utilities. Intuitively, PE means that in order to improve the utility of one agent, we have to decrease the utility of another.

Definition (Pareto-efficiency). An allocation x is Pareto-dominated by x' if $\widetilde{u}_i(x'_i) \geq \widetilde{u}_i(x_i)$ for all $i \in [n]$ and $\widetilde{u}_j(x'_j) > \widetilde{u}_j(x_j)$ for some $j \in [n]$. An allocation x is Pareto-efficient if no other allocation Pareto-dominates x.

Exercise: Relate this to the definition of PE in one-sided matching. *(try at home)*

Theorem. Let us assume that, for each agent i, the utility is strictly increasing up to the demand d_i. That is, $\widetilde{u}_i(z) < \widetilde{u}_i(z')$ for all $z < z' \leq d_i$, and $\widetilde{u}_i(z) = \widetilde{u}_i(d_i)$ for all $z \geq d_i$. Then, an allocation x is Pareto-efficient if and only if $\ell(x) = 0$.

Zero resource loss and Pareto-efficiency (cont'd)

Proof. We will prove the "if" direction in class. You will prove the "only if" direction in the homework.

Suppose $\ell(x) = 0$. We will show that x is Pareto-efficient. First note,

$$\ell(x) = 0 \qquad \Longleftrightarrow \qquad \ell^{\mathrm{ud}}(x) = 0 \quad \text{or} \quad \ell^{\mathrm{or}}(x) + \ell^{\mathrm{ur}}(x) = 0.$$

Say $\ell^{\mathrm{ud}}(x) = 0$. This implies that $x_i \geq d_i$ for all $i \in [n]$, *i.e.*, we have allocated at least the demand to all agents. As the utility does not increase after the demand, we have that every agent has achieved their maximum possible utility. As no agent can improve their utility, x is Pareto-efficient.

Recall, resource loss. The resource loss $\ell(x)$ is the amount of idle resources (either over- or un-allocated) when there is an unmet demand:

$$\ell(x) = \min\left(\ell^{\mathrm{ud}}(x),\ \ell^{\mathrm{or}}(x) + \ell^{\mathrm{ur}}(x)\right) \qquad \text{where,}$$

$$\ell^{\mathrm{ur}}(x) = 1 - \sum_{i=1}^{n} x_i, \qquad \ell^{\mathrm{or}}(x) = \sum_{i=1}^{n} (x_i - d_i)^+, \qquad \ell^{\mathrm{ud}}(x) = \sum_{i=1}^{n} (d_i - x_i)^+.$$

Zero resource loss and Pareto-efficiency (cont'd)

Now say $\ell^{\mathrm{or}}(x) + \ell^{\mathrm{ur}}(x) = 0$. This implies,

$$\sum_{i=1}^{n} x_i = 1, \qquad \text{and} \qquad x_i \leq d_i, \ \forall\, i \in [n].$$

This means, to increase j's utility (for whom $x_j < d_j$), we will need to take resources away from some other agent k. But as the utility is strictly increasing up to the demand d_k, this will decrease k's utility. Hence x is Pareto-efficient. □

Next, we will introduce the Max-min fairness (MMF) algorithm (mechanism) which satisfies all three desiderata.

Max-min fairness: example

We will start with following example with 4 agents and $e_i = 1/4$ for all i. Here, the fair share is the $r\frac{e_i}{e}$ quantity in the algorithm (next slide).

	Agent 1	Agent 2	Agent 3	Agent 4
Demands	0.5	.26	.10	.80
Round 1: fair share $r\frac{e_i}{e}$	.25	.25	.25	.25
allocation			**0.1**	
Round 2: fair share $r\frac{e_i}{e}$	.3	.3	–	.3
allocation		**.26**	–	
Round 3: fair share $r\frac{e_i}{e}$	.32	–	–	.32
allocation	**.32**	–	–	**.32**
Final allocation	**.32**	**.26**	**.1**	**.32**

The Max-min fairness (MMF) algorithm

1. **Given:** endowments $\{e_i\}_{i\in[n]}$, bids $\{b_i\}_{i\in[n]}$.
2. $r \leftarrow 1$, $e \leftarrow 1$ `# amount of resources/endowment left`
3. $S \leftarrow [n]$ `# unallocated agents`
4. $x_i \leftarrow \texttt{NULL}$ for all $i \in [n]$. `# final allocation`
5. **For** j in ascending order of b_i/e_i:
6.   **If** $b_j \leq r\frac{e_j}{e}$: `# if an agent's demand is less than her 'fair share'`
7. $x_j \leftarrow b_j$, `# allocate her demand`
8. $S \leftarrow S\backslash\{j\}$, $\quad r \leftarrow r - b_j$, $\quad e \leftarrow e - e_j$. `# update` S, r, e
9.   **Else** :
10. $x \leftarrow r\frac{e_k}{e}$ for all $k \in \mathcal{S}$; `# allocate remaining resources`
11.   **break** `# exit`
12. return x

Properties of MMF

Theorem. MMF is DSIC, IR, and efficient.

Proof (efficiency). Let us recall the definition of efficiency.

Recall: A mechanism is efficient if $\ell(x(d; M)) = 0$, where

$$\ell(x) = \min\left(\ell^{\mathrm{ud}}(x),\ \ell^{\mathrm{or}}(x) + \ell^{\mathrm{ur}}(x)\right)$$

$$\ell^{\mathrm{ur}}(x) = 1 - \sum_{i=1}^{n} x_i, \qquad \ell^{\mathrm{or}}(x) = \sum_{i=1}^{n}(x_i - d_i)^+, \qquad \ell^{\mathrm{ud}}(x) = \sum_{i=1}^{n}(d_i - x_i)^+.$$

That is, when agents are truthful, there are no idle resources when there is an unmet demand.

Let x be the allocation by MMF when all agents have truthfully reported $b_j = d_j$. We will first show the following claim.

Claim: MMF never allocates more than d_i to any agent.

Proof of efficiency (cont'd)

Proof of claim: Allocations occur either in line 7 or line 10.

- In line 7: an agent is allocated exactly her demand.

- In line 10: when the Else condition is triggered, $b_j > r\frac{e_j}{e}$ for all remaining agents who are allocated in line 10 (Why?).

- Hence, $x_j = \frac{e_j}{e} r < b_j = d_j$. □

(Continuing with proof of efficiency) By the claim, $\ell^{\mathrm{or}}(x) = \sum_{i=1}^{n}(x_i - d_i)^+ = 0$. Now if, $\ell^{\mathrm{ur}}(x) = 0$, then $\ell(x) = 0$ and we are done.

So say $\ell^{\mathrm{ur}}(x) > 0$.

$\implies$ MMF never entered line 10 (in line 10, we allocate all remaining resources).

$\implies$ All allocations were in line 7, where we set $x_i = b_i = d_i$.

$\implies$ We have $\ell^{\mathrm{ud}}(x) = \sum_{i=1}^{n}(d_i - x_i)^+ = 0$.

$\implies$ $\ell(x) = 0$. □

Proof of IR

Recall, IR: . A truthful agent's utility is at least as much when using her endowment, *i.e.*, $u_i(d_i, b_{-i}; M) \geq \widetilde{u}_i(e_i)$, for all $b_{-i} \in \mathbb{R}_+^{n-1}$.

Proof (IR). Consider agent i and assume $b_i = d_i$ (but not necessarily $b_j = d_j$ for $j \neq i$). If agent i was allocated in line 7, then

$$x_i = b_i = d_i \qquad \Longrightarrow \qquad u_i(d_i, b_{-i}; M^{\mathrm{MMF}}) = \widetilde{u}(x_i) = \widetilde{u}(d_i) \geq \widetilde{u}(e_i)$$

as she cannot achieve her utility than $\widetilde{u}(d_i)$.

Suppose then that the agent was allocated in line 10. The following calculations show that the quantity $\frac{r}{e}$ is non-decreasing after each pass of the **For** loop:

$$\texttt{end of loop:} \quad \frac{r - b_j}{e - e_j} = \frac{r(1 - b_j/r)}{e(1 - e_j/e)} \overset{(a)}{>} \frac{r(1 - e_j/e)}{e(1 - e_j/e)} = \frac{r}{e} \quad \texttt{:beginning of loop}$$

Here, (a) uses the fact that $b_j < \frac{r}{e} e_j$. As $\frac{r}{e} = 1$ at the beginning, for all allocations in line 10, we have $x_j = \frac{r}{e} e_j \geq e_j$, and hence $\widetilde{u}(x_i) \geq \widetilde{u}(e_i)$. □

Proof of DSIC

Recall, DSIC. No agent should have an incentive to misreport her demand regardless of others' bids, *i.e.*, $u_i(d_i, b_{-i}; M) \geq u_i(b_i, b_{-i}; M)$ for all $b_i \in \mathbb{R}_+, b_{-i} \in \mathbb{R}_+^{n-1}$.

Proof (DSIC). Fix the bids b_{-i} of the other agents. Let x be the allocation when agent i truthfully reports $b_i = d_i$ and x' be the allocation when she reports some other $b_i \neq d_i$.

We will show:

(i) Under-stating the demand can hurt: If $b_i < d_i$, then $\widetilde{u}_i(x_i') \leq \widetilde{u}_i(x_i)$.

(ii) Over-stating the demand does not help: If $b_i > d_i$, then $\widetilde{u}_i(x_i') = \widetilde{u}_i(x_i)$.

We will prove *(i)* in class. You will prove *(ii)* in the homework.

Proof of DSIC (cont'd)

We begin with the following observation: requesting an arbitrarily large number of resources does not increase one's allocation under MMF. For instance, in the example on page 46, Agent 1 receives 0.32 resources for any request larger than 0.32. We state this formally below.

Observation. Fix the bids b_{-i} of the others. Then, there exists some value $\widetilde{x}_i$ such that, for all $b_i \geq \widetilde{x}_i$, her allocation will be $\widetilde{x}_i$.

Proof. The agent was allocated $\widetilde{x}_i = \frac{r}{e} e_i$ and r, e only depend on the bids of the others. For any $b_i \geq \frac{r}{e} e_i$, the **Else** condition will be triggered with the same set of agents receiving the requested bids b_j in line 7 (with the remaining set of agents receiving $\frac{r}{e} e_j$ in line 10).

Intuitively, this observation says that the maximum amount of resources an agent can receive depends only on b_{-i} (other's bids) and not on her own bid!

Proof of DSIC (cont'd)

Proof of (i). If $b_i < d_i$, then $\widetilde{u}_i(x_i') \leq \widetilde{u}_i(x_i)$

When an agent reports more, they get pushed back in the ascending order of b_j/e_j. So there could be three cases when $b_i < d_i$:

Case 1: The agent is allocated $x_i = d_i$ when reporting truthfully, and $x_i' = b_i$ when under-reporting, both in line 7. Then, $\widetilde{u}_i(x_i') < \widetilde{u}_i(x_i)$.

Case 2: The agent is allocated $x_i = \frac{r}{e} e_i$ in line 10 when reporting b_i, where r, e depends on other's reports. Then, by our observation above, the agent is still allocated $x_i = \frac{r}{e} e_i$ even when reporting truthfully. Hence, $x_i = x_i' = \frac{r}{e} e_i$, and $\widetilde{u}_i(x_i') = \widetilde{u}_i(x_i)$.

Case 3: The agent is allocated $x_i' = b_i$ in line 7 when under-reporting b_i, and x_i in line 10 when reporting truthfully. Then,

$$x_i' = b_i \overset{(a)}{<} \frac{r'}{e'} e_i \overset{(b)}{\leq} \frac{r''}{e''} e_i = x_i < d_i.$$

Proof of DSIC (cont'd)

Here, in (a), r', e' are the values of r, e when the for loop reaches agent i, when i reports b_i. By the condition for the for loop, $b_i < \frac{r'}{e'} e_i$. Next, in (b), r'', e'' are the values of r, e when the **Else** condition is triggered when i reports d_i; (b) follows from the fact that $\frac{r}{e}$ is non-decreasing (we showed this in the IR proof).
As $x_i' < x_i < d_i$, we have $\widetilde{u}(x_i') < \widetilde{u}(x_i)$. □

This proves *(i)*. You will prove *(ii)* in the homework.

Summary: Mechanism design without money

Recall the following three challenges we introduced at the outset:

1. **Incentive compatibility:** An agent's utility depends on some *private* information that the mechanism designer does not observe. A mechanism is *incentive compatible* if every agent finds it optimal to report their true information.
2. **Individual rationality:** Ensure that agents are better off participating in the mechanism than not participating.
3. **Efficiency/Stability:** Ensure that when self-interested agents play the game, the desired social goal is achieved.

Describe/formulate these challenges in the three problems we studied.

Summary: Mechanism design without money

Problem	IC	IR	Efficiency/Stability
Stable matching	Truthfully report preferences over the other side of the market	–	Produce a stable matching
One-sided matching	Truthfully report preferences over others' goods	Each agent's allocation is no less preferred than her own initial good	Pareto-optimal allocations
Fair resource allocation	Truthfully report demands	Each agent does at least as well as using her endowment	No idle resources when there is unmet demand