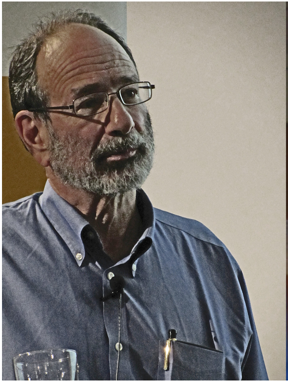
SF2972: Game theory

Introduction to matching

The 2012 Nobel Memorial Prize in Economic Sciences:
awarded to Alvin E. Roth and Lloyd S. Shapley for "the theory of stable allocations and the practice of market design"



The related branch of game theory is often referred to as **matching theory**, which studies the design and performance of platforms for transactions between agents. Roughly speaking, it studies who interacts with whom, and how: which applicant gets which job, which students go to which universities, which donors give organs to which patients, and so on.

Plan

Many methods for finding desirable allocations in matching problems are variants of two algorithms:

1. The deferred acceptance algorithm
2. The top trading cycle algorithm

For each of the two algorithms, I will do the following:

- State the algorithm.
- State and prove nice properties of outcomes generated by the algorithm.
- Solve an example using the algorithm.
- Describe application(s).
- Give you related homework exercises.

The deferred acceptance (DA) algorithm: reference

- D. Gale and L.S. Shapley, 1962, College Admissions and the Stability of Marriage. *American Mathematical Monthly* 69, 9–15.
- Only seven pages. . .
- . . . and, yes, stability of marriage!

The deferred acceptance (DA) algorithm: marriage problem

- Men and women have strict preferences over partners of the opposite sex
  - You may prefer staying single to marrying a certain partner
- A *match* is a set of pairs of the form (m, w), (m, m), or (w, w) such that each person has exactly one partner.
- Person i is *unmatched* if the match includes (i, i).
- i is *acceptable* to j if j prefers i to being unmatched.
- Given a proposed match, a pair (m, w) is *blocking* if both prefer each other to the person they're matched with.
  - m prefers w to his match-partner
  - w prefers m to her match-partner
- A match is *unstable* if someone has an unacceptable partner or if there is a blocking pair. Otherwise, it is *stable*.
- A match is *man-optimal* if it is stable and there is no other stable match that some man prefers. Woman-optimal analogously.

The deferred acceptance (DA) algorithm: statement

Input: A nonempty, finite set M of men and W of women. Each man (woman) ranks acceptable women (men) from best to worst.

DA algorithm, men proposing:

1. Men are not allowed to propose to women that find them unacceptable. Other than that:
2. Each man proposes to the highest ranked woman on his list.
3. Women hold at most one offer (her most preferred acceptable proposer), rejecting all others.
4. Each rejected man removes the rejecting woman from his list.
5. If there are no new rejections, stop. Otherwise, iterate.
6. After stopping, implement proposals that have not been rejected.

Remarks:

1. DA algorithm, women proposing: switch roles!
2. Deferred acceptance: receiving side defers final acceptance of proposals until the very end.

The deferred acceptance (DA) algorithm: nice properties

1. The deferred acceptance algorithm ends with a stable match.
   - Ends: the set of women a man can propose to does not increase and decreases for at least one (rejected) man.
   - By construction, no person is matched to an unacceptable candidate.
   - No (m, w) can be a blocking pair: if m strictly prefers w to his current match, he must have proposed to her and been rejected in favor of a candidate that w liked better. As the algorithm goes on, w can only do better. So w finds her match better than m.
2. This match is man-optimal (woman-pessimal).
3. Men have no incentives to lie about their preferences, women might.
   - Strategy-proof for men
   - See homework exercise
4. There is no mechanism that always ends in a stable match and that is strategy-proof for all participants.

Of course, similar results apply if women propose.

The deferred acceptance (DA) algorithm: example

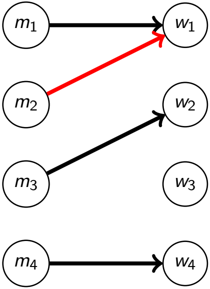
- For convenience $|M| = |W| = 4$.
- All partners of opposite sex are acceptable.
- Ranking matrix:

$$\begin{array}{c|cccc|}
 & w_1 & w_2 & w_3 & w_4 \\
\hline
m_1 & 1,3 & 2,3 & 3,2 & 4,3 \\
m_2 & 1,4 & 4,1 & 3,3 & 2,2 \\
m_3 & 2,2 & 1,4 & 3,4 & 4,1 \\
m_4 & 4,1 & 2,2 & 3,1 & 1,4 \\
\hline
\end{array}$$

- Interpretation: entry $(1,3)$ in the first row and first column indicates that m_1 ranks w_1 first among the women and that w_1 ranks m_1 third among the men.

The deferred acceptance (DA) algorithm: example

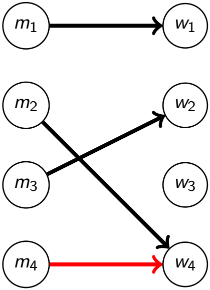
	w_1	w_2	w_3	w_4
m_1	1, 3	2, 3	3, 2	4, 3
m_2	1, 4	4, 1	3, 3	2, 2
m_3	2, 2	1, 4	3, 4	4, 1
m_4	4, 1	2, 2	3, 1	1, 4



w_1 is the only person to receive multiple proposals; she compares m_1 (rank 3) with m_2 (rank 4) and rejects m_2. Strike this entry from the matrix and iterate.

The deferred acceptance (DA) algorithm: example

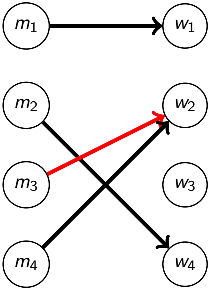
	w_1	w_2	w_3	w_4
m_1	1, 3	2, 3	3, 2	4, 3
m_2	1, 4	4, 1	3, 3	2, 2
m_3	2, 2	1, 4	3, 4	4, 1
m_4	4, 1	2, 2	3, 1	1, 4



w_4 is the only person to receive multiple proposals; she compares m_2 (rank 2) with m_4 (rank 4) and rejects m_4. Strike this entry from the matrix and iterate.

The deferred acceptance (DA) algorithm: example

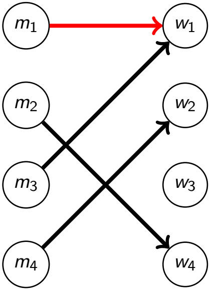
	w_1	w_2	w_3	w_4
m_1	1, 3	2, 3	3, 2	4, 3
m_2	1, 4	4, 1	3, 3	2, 2
m_3	2, 2	1, 4	3, 4	4, 1
m_4	4, 1	2, 2	3, 1	1, 4



w_2 is the only person to receive multiple proposals; she compares m_3 (rank 4) with m_4 (rank 2) and rejects m_3. Strike this entry from the matrix and iterate.

The deferred acceptance (DA) algorithm: example

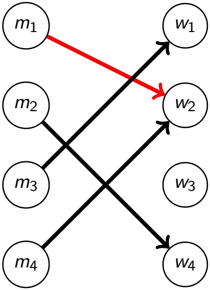
	w_1	w_2	w_3	w_4
m_1	1, 3	2, 3	3, 2	4, 3
m_2	1, 4	4, 1	3, 3	2, 2
m_3	2, 2	1, 4	3, 4	4, 1
m_4	4, 1	2, 2	3, 1	1, 4



w_1 is the only person to receive multiple proposals; she compares m_1 (rank 3) with m_3 (rank 2) and rejects m_1. Strike this entry from the matrix and iterate.

The deferred acceptance (DA) algorithm: example

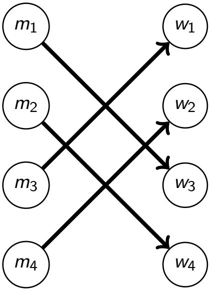
	w_1	w_2	w_3	w_4
m_1	1, 3	2, 3	3, 2	4, 3
m_2	1, 4	4, 1	3, 3	2, 2
m_3	2, 2	1, 4	3, 4	4, 1
m_4	4, 1	2, 2	3, 1	1, 4



w_2 is the only person to receive multiple proposals; she compares m_1 (rank 3) with m_4 (rank 2) and rejects m_1. Strike this entry from the matrix and iterate.

The deferred acceptance (DA) algorithm: example

	w_1	w_2	w_3	w_4
m_1	1, 3	2, 3	3, 2	4, 3
m_2	1, 4	4, 1	3, 3	2, 2
m_3	2, 2	1, 4	3, 4	4, 1
m_4	4, 1	2, 2	3, 1	1, 4



No rejections; the algorithm stops with stable match

$$(m_1, w_3), (m_2, w_4), (m_3, w_1), (m_4, w_2).$$

DA with men proposing: optimal for men

We claimed that the men-proposing DA algorithm ends with a stable matching that is optimal for men: each man is at least as well off under this match as under any other stable matching. (Analogously, it is the worst/'pessimal' stable matching for women). Why?

Proof strategy:

- Call a woman 'possible' for a man if they are partners in some stable matching.
- By induction on the stages of the DA algorithm, show that in each round, men are only rejected by impossible women.
- Since men propose to women in order of preference, each man ends up with his first/best possible partner.

Proof

- Assume: at no earlier stage has a man been rejected by a woman that is possible.
- At the current stage, suppose woman w rejects man m in favor of man m'. Then w is impossible for m:

  Suppose we try to find a stable match pairing m to w. Then m' must be matched to someone else.

  - Not with a woman better than w: since m' made an offer to w, all women he likes more must have refused him. By induction, these are impossible for him!
  - Not with a woman worse than w: then m' and w would elope as they would rather be with each other.

A marriage problem may have several stable matchings. However:

Theorem (Rural hospital theorem)

The set of men and women who remain single is the same in every stable matching.

This is called the ‘rural hospital theorem’: in the US, the National Resident Matching Program uses deferred acceptance to match physicians with hospitals. Since hospitals in sparsely populated areas had trouble filling their positions, they wondered whether changing the algorithm to some other stable matching would help them fill the empty spots.

Proof

- Let M^{DA}, W^{DA} be the sets of men and women matched in the man-optimal (woman-pessimal) stable matching: not-single.
- Let M', W' be the sets of men and women matched in another stable matching.
- Any man in M' must also be matched in the man-optimal stable matching: $M' \subseteq M^{DA}$ and $|M'| \leq |M^{DA}|$.
- Any woman matched in the woman-pessimal stable matching must also be matched in W': $W^{DA} \subseteq W'$ and $|W^{DA}| \leq |W'|$.
- Also, $|M^{DA}| = |W^{DA}|$ and $|M'| = |W'|$.
- So $|M'| = |M^{DA}| = |W^{DA}| = |W'|$.
- So $M' = M^{DA}$ and $W' = W^{DA}$.

DA with men proposing: men shouldn't lie

In the men-proposing DA algorithm, men cannot benefit from lying about their preferences. Why?

Proof strategy:

- Fix reported preferences of all women and all but one man.
- Show that any profile of reported preferences for this man can be weakly improved upon by a sequence of changes ending with a truthful report.

Proof

Suppose the man m states preferences leading to a match μ that pairs him to woman w (if he stays single, that is worst anyway, so truthful reporting cannot harm). The following changes weakly improve upon the result:

- Report that w is his only acceptable candidate.
  Match μ remains stable (Only m changed preferences; earlier w and possibly others were acceptable, now only w, so there are fewer candidates for blocking pairs.) By the rural hospital theorem, m must be matched, and so must be paired with w.
- Report truthfully, but truncate at w.
  DA cannot result in a match where m is single: such a matching was blocked in the previous case and with m's new preferences there are even more candidates for blocking pairs.
- Report truthfully.
  This won't change the DA outcome, since convergence in the previous step was independent of what the man could have said after w.

Homework exercise 1

Consider the ranking matrix

$$\begin{array}{c|ccccc|}
 & w_1 & w_2 & w_3 & w_4 & w_5 \\
\hline
m_1 & 1,3 & 3,1 & 2,1 & 5,1 & 4,2 \\
m_2 & 1,2 & 2,2 & x,x & 3,2 & 4,1 \\
m_3 & 3,1 & 2,3 & 1,2 & x,x & 4,x \\
\hline
\end{array}$$

Here, an x indicates an unacceptable partner. For instance, m_3 ranks w_5 fourth, but w_5 would rather be single than be matched with m_3.

(a) Find a stable matching using the men-proposing DA algorithm.

(b) Find a stable matching using the women-proposing DA algorithm.

(c) Are there other stable matchings?

Homework exercise 2

Consider the ranking matrix

$$\begin{array}{c|cc|} & w_1 & w_2 \\ \hline m_1 & 1,2 & 2,1 \\ m_2 & 2,1 & 1,2 \\ \hline \end{array}$$

(a) Find a stable matching using the men-proposing DA algorithm.

(b) Find a stable matching using the women-proposing DA algorithm.

(c) Suppose that w_1 lies about her preferences and says that she only finds m_2 acceptable. What is the outcome of the men-proposing DA algorithm now? Verify that both women are better off than under (a): it may pay for the women to lie!

Prove: if a mechanism always picks a stable matching (given reported preferences), then in the marriage problem of the previous exercise there is always some agent who can benefit from lying about his or her preferences.

Hint: Verify:

- The matchings you found in (a) and (b) are the only stable ones.
- If the former is chosen, let w_1 lie as in (c) and show that the resulting problem has only one stable matching, which is better for w_1.
- Analogously (note the problem's symmetry), if the latter is chosen, m_1 can profitably lie about his preferences.

The college admissions problem

The original motivation for the paper of Gale and Shapley.

- n applicants, m colleges, q_i the quota of college i
- each applicant strictly ranks (no ties!) colleges, each college strictly ranks applicants
- as in the marriage problem, applicants may leave out unacceptable colleges and vice versa
- a matching of applicants to colleges can be *blocked* by an applicant a and a college c if
  - a prefers c to her current match
  - c prefers a to one of its current matches
- a matching is *stable* if all matched applicants and colleges find each other acceptable and there is no blocking pair.
- a stable assignment is (student) optimal if each applicant is at least as well off under it as under any other stable assignment.

Deferred acceptance for college admission

1. If a college finds a student unacceptable, the student is not allowed to apply there.
2. First, all students apply to the college of their first choice.
3. A college with quota q puts on a waiting list the q applicants it ranks highest, or all applicants if there are fewer than q, and rejects the rest.
4. Rejected applicants apply to their second choice and again each college keeps the (at most) q favorite applicants on its waiting list and rejects the rest.
5. The algorithm terminates when every applicant is either on a waiting list or has been rejected by every college to which (s)he is willing and permitted to apply.
6. At this stage, the colleges accept the students on their waiting list.

Stability and optimality in college admission

This deferred acceptance algorithm gives an allocation that is both stable and optimal:

1. Stability follows as in the marriage problem: if applicant α would rather go to college C, then either that college does not allow α to apply, or it must have rejected α because the candidates on the waiting list are better.
2. Optimality: each applicant is at least as well off under the assignment given by the deferred acceptance algorithm as under any other stable assignment. (Proof as before in the marriage problem)

Roommate problem: stable matchings don't always exist

In the marriage and college admissions problem, stable matchings always existed. This is not necessarily the case in other matching problems. Gale and Shapley illustrate this with a 'roommate problem'.

A group of students must be matched in pairs to share dormitory rooms.

Example: Four students: α, β, γ, and δ.
α ranks β first,
β ranks γ first,
γ ranks α first,
all three rank δ last.
Regardless of δ's preferences, there is no stable matching: whoever shares a room with δ wants to change and can find an eager friend among the other two.

The top trading cycle (TTC) algorithm: reference

- L.S. Shapley and H. Scarf, 1974, On Cores and Indivisibility. *Journal of Mathematical Economics* 1, 23–37.
- The algorithm is described in section 6, p. 30, and attributed to David Gale.

The top trading cycle (TTC) algorithm: statement

Input: Each of $n \in \mathbb{N}$ agents owns an indivisible good (a house) and has strict preferences over all houses.
Convention: agent i initially owns house h_i.
Question: Can the agents benefit from swapping houses?
TTC algorithm:

1. Each agent i points to her most preferred house (possibly i's own); each house points back to its owner.
2. This creates a directed graph. In this graph, identify cycles.
   - Finite: cycle exists.
   - Strict preferences: each agent is in at most one cycle.
3. Give each agent in a cycle the house she points at and remove her from the market with her assigned house.
4. If unmatched agents/houses remain, iterate.

The top trading cycle (TTC) algorithm: nice properties

- The TTC assignment is such that no subset of owners can make all of its members better off by exchanging the houses they initially own in a different way.
  - Suppose a subset S can make all its members better off.
  - S contains no members from the first cycle in TTC: those already get their favorite.
  - S contains no members from the second cycle in TTC: they get the best of what's left after round 1, so making them better off requires a member of the first cycle, but we already ruled out that those were in S.
  - Etc.

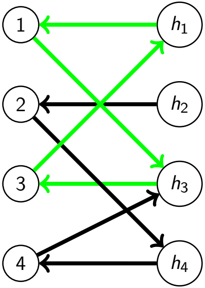
The top trading cycle (TTC) algorithm: nice properties

- It is never advantageous to an agent to lie about preferences if the TTC algorithm is used.
  - Consider an agent who reports truthfully and gets a house in round t.
  - No change in the report can give the agent a house that was assigned in earlier rounds (those cycles remain, no matter what the agent says).
  - And the agent gets the best of what's left by reporting truthfully.

The top trading cycle (TTC) algorithm: example

Agents' ranking from best (left) to worst (right):

1 : (h_3, h_2, h_4, h_1)
2 : (h_4, h_1, h_2, h_3)
3 : (h_1, h_4, h_3, h_2)
4 : (h_3, h_2, h_1, h_4)

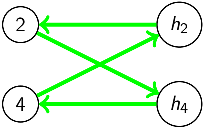


- Cycle: $(1, h_3, 3, h_1, 1)$.
- So: 1 get h_3 and 3 gets h_1. Remove them and iterate.

Only agents 2 and 4 left with updated preferences:

$2 : \quad (h_4, h_2)$

$4 : \quad (h_2, h_4)$



- Cycle: $(2, h_4, 4, h_2, 2)$.
- So: 2 gets h_4 and 4 gets h_2. Done!
- Final match:

$$(1, h_3), (2, h_4), (3, h_1), (4, h_2).$$

The top trading cycle (TTC) algorithm: application 1

- A. Abdulkadiroğlu and T. Sönmez, 2003. School Choice: A Mechanism Design Approach. *American Economic Review* 93, 729–747.
- How to assign children to schools subject to priorities for siblings and distance?

Input:

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

Modified TTC algorithm:

1. Each remaining student points at her most preferred unfilled school; each unfilled school points at its most preferred remaining student.
2. Cycles are identified and students in cycles are matched to the school they point at.
3. Remove assigned students and full schools.
4. If unmatched students remain, iterate.

- A.E. Roth, T. Sönmez, M.U. Ünver, 2004. Kidney Exchange. *Quarterly Journal of Economics* 119, 457–488.
- A case with patient-donor pairs: a patient in need of a kidney and a donor (family, friend) who is willing to donate one.
- Complications arise due to incompatibility (blood/tissue) groups, etc.
- So look at trading cycles: patient 1 might get the kidney of donor 2, if patient 2 gets the kidney of donor 1, etc.

The top trading cycle (TTC) algorithm: homework exercise 4

Apply the TTC algorithm to the following case:

$$\begin{array}{ll}
1: & (h_5, h_2, h_1, h_3, h_4) \\
2: & (h_5, h_4, h_3, h_1, h_2) \\
3: & (h_4, h_2, h_3, h_5, h_1) \\
4: & (h_2, h_1, h_5, h_3, h_4) \\
5: & (h_2, h_4, h_1, h_5, h_3)
\end{array}$$