Agent-Based Systems

Paolo Turrini

www.dcs.warwick.ac.uk/~pturrini p.turrini@warwick.ac.uk

Stable Matching

...and stable marriages

Plan for Today

Matching deals with scenarios where agents have preferences over what other agent they get paired up with. Important applications:

- Matching junior doctors to hospitals
- Matching school children to schools
- Kidney exchanges (different model from what we'll discuss)

Today is going to be an introduction to this topic, largely focusing on the basic scenario of **one-to-one** matching:

- **the stable marriage problem** and the connection to coalitional games
- finding a **stable** solution with the **deferred-acceptance algorithm**
- various **extensions** of the model and other **properties** of solutions

A good general reference, somewhat emphasising algorithmic issues, is the chapter by Klaus et al. (2016).

B. Klaus, D.F. Manlove, and F. Rossi.
Matching under Preferences.
In: *Handbook of Computational Social Choice*, Cambridge University Press, 2016.

We are given:

- *n* **men** and *n* **women**
- each has a strict **preference** order over members of the opposite sex [1]

We seek:

- a **stable** matching of men to women: no man and woman should want to divorce their assigned partners and run off with each other

[1] Less traditional variants come with fewer constraints.

A **stable marriage problem** is a tuple $\langle M, W, \succ^{\boldsymbol{m}}, \succ^{\boldsymbol{w}} \rangle$, where

- M is a finite set of **men**, W a finite set of **women**, $|M| = |W| = \mathbf{n}$,
- $\succ^{\boldsymbol{m}} = (\succ^m_1, \ldots, \succ^m_n)$ is a profile of strict **preference orders** on W, one for each man $i \in M$, and

- $\succ^{\boldsymbol{w}} = (\succ^w_1, \ldots, \succ^w_n)$ is a profile of strict **preference orders** on M, one for each woman $i \in W$.

Exercise: *What is 'stability' in matching?*

The Gale-Shapley Algorithm

Theorem (Gale and Shapley, 1962)

There exists a stable matching for any combination of preferences of men and women.

Proof: Consider the **deferred-acceptance algorithm** below.

- In each round, every man who is not yet engaged proposes to his favourite amongst the women he has not yet proposed to.
- In each round, every woman picks her favourite from the proposals she's receiving and the man she's currently engaged to (if any).
- Stop when everyone is engaged.

We observe: First, this always **terminates** with a complete matching. Second, that matching must be **stable**: for if not, that unhappy man would have proposed to that unhappy woman . . . ✓

D. Gale and L.S. Shapley.
College Admissions and the Stability of Marriage.
American Mathematical Monthly, 69:9–15, 1962.

Exercise: Number of Rounds

Recall that n is the number of men (and also the number of women).
How many **rounds** *does it take for the algorithm to terminate and how many* **proposals** *will be made in the process? Best case? Worst case?*

M-Optimal and W-Optimal Matchings

A stable matching is **M-optimal** (W-optimal) if every **man** (woman) likes it at least as much as any other stable matching.

Theorem (Gale and Shapley, 1962)

The matching returned by the **deferred-acceptance algorithm** *(with men proposing) is* **M-optimal**.

Proof: For the sake of contradiction, suppose m is matched with w but prefers w' he'd be matched to in some other stable matching.
So m proposed to w' before w, but w' rejected. W.l.o.g., assume this was the first rejection of a "stable partner".

Let m' be the man engaged to w' at the time of rejection. Thus, m' prefers w' over all stable partners (because no woman previously rejected a stable partner). Hence, that other matching cannot be stable, as m' and w' would prefer to run off together. ✓
Remark: One can also show that the outcome is always W-pessimal.

Fairness

M-optimal matchings (returned by the deferred-acceptance algorithm) arguably are not fair. But what is **fair**?

- One option is to implement the stable matching that **minimises** the **regret** of the person worst off (regret = number of members of the opposite sex they prefer to their assigned partner).

  Gusfield (1987) gives an algorithm for min-regret stable matchings.

- Similarly, we can implement the stable matching that maximises **average satisfaction** (i.e., that minimises average regret).

  Irving et al. (1987) give an algorithm for this problem.

D. Gusfield.
Three Fast Algorithms for Four Problems in Stable Marriage.
SIAM Journal of Computing, 16(1):111–128, 1987.

R.W. Irving, P. Leather, and D. Gusfield.
An Efficient Algorithm for the "Optimal" Stable Marriage.
Journal of the ACM, 34(3):532–543, 1987.

Stable Marriages under Incomplete Preferences

In an important generalisation of the simple stable marriage problem, people are allowed to specify which members of the opposite sex they consider **acceptable**, and they only report a strict ranking of those.

- Now the assumption is that a man/woman would rather remain **single** than marry a partner they consider unacceptable.
- Now a matching is **stable** if no couple has an incentive to run off together and if no individual has an incentive to leave their assigned partner and be single.
- The **deferred-acceptance algorithm** can easily be extended to this setting: simply stipulate that men don't propose to unacceptable women and women don't accept unacceptable men.

This is called the stable marriage problem with **incomplete preferences**.

Impossibility of Strategyproof Stable Matching

A matching mechanism is **strategyproof** if it never gives either a man or a woman an incentive to misrepresent their preferences.

Theorem (Roth, 1982)

There exists no matching mechanism that is **stable** *as well as* **strategyproof** *for both men and women.*

The proof on the next slide uses only two men and two women, but it relies on a manipulation involving agents misrepresenting which partners they find *acceptable*. Alternative proofs, using three men and three women, involve only changes in preference (not acceptability).

A.E. Roth.
The Economics of Matching: Stability and Incentives.
Mathematics of Operations Research, 7:617–628, 1982.

Proof

Suppose there are two men and two women with these preferences:

$$
\begin{array}{ll}
m_1 : w_1 \succ w_2 \mid & m_2 : w_2 \succ w_1 \mid \\
w_1 : m_2 \succ m_1 \mid & w_2 : m_1 \succ m_2 \mid
\end{array}
$$

$\Rightarrow$ 2 stable matchings: $\{(m_1, w_1), (m_2, w_2)\}$ and $\{(m_1, w_2), (m_2, w_1)\}$ So any stable mechanism will have to pick one of them.

- Suppose the mechanism were to pick $\{(m_1, w_1), (m_2, w_2)\}$.
  Then w_2 can pretend that she finds m_2 unacceptable, thereby making $\{(m_1, w_2), (m_2, w_1)\}$ the only stable matching.
- Suppose the mechanism were to pick $\{(m_1, w_2), (m_2, w_1)\}$.
  Then m_1 can pretend that he finds w_2 unacceptable, thereby making $\{(m_1, w_1), (m_2, w_2)\}$ the only stable matching.

Hence, for any possible stable matching mechanism there is a situation where someone has an incentive to manipulate. ✓

Preferences with Ties

We can further generalise the stable marriage problem by also allowing for **ties**, i.e., by allowing each agent to have a weak preference order over (acceptable) members of the opposite sex.

We can still **compute a stable matching** in polynomial time:

- arbitrarily break the ties (i.e, refine weak into strict orders)
- apply the standard deferred-acceptance algorithm

Now (first time today) different stable matchings can **differ in size**:

$$
\begin{array}{ll}
m_1 : w_1 \mid w_2 & \quad m_2 : w_1 \succ w_2 \\
w_1 : m_1 \sim m_2 & \quad w_2 : m_2 \mid m_1
\end{array}
$$

Both $\{(m_2, w_1)\}$ and $\{(m_1, w_1), (m_2, w_2)\}$ are stable.
Finding a **maximal** stable matching is NP-hard (Manlove et al., 2002).

D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita.
Hard Variants of Stable Marriage.
Theoretical Computer Science, 276(1–2):261–279, 2002.

Variants

Variants and generalisations are applicable to many scenarios:

- **Residents-Hospitals Problem**
  - Matching of junior doctors (residents) to hospitals.
  - Many-to-one variant of stable marriage problem with incomplete preferences, with each hospital having a certain capacity.

- **School Choice**
  - Matching of school children to schools.
  - Similar, but schools have **priorities** rather than preferences (distance to home, sibling already at school, etc.).

Main difference is interpretational: schools are not economic agents.

When hospitals/schools have weak preferences/priorities, we need to find a way to **break ties** when capacity limits are reached.

Case Study: Amsterdam School Choice

System used prior to 2015 ("adaptive Boston mechanism"):

- Use lottery to rank all children. Use ranking to refine every school's priority list into a strict order.
- Ask children to announce their top choices. Award top choices subject to capacity constraints and following refined priority lists. Remove matched children and places from system. Repeat.

System introduced for 2015 (D-A with local tie-breaking):

- Refine priorities as above, but use separate lottery for each school.
- Use many-to-one variant of deferred-acceptance algorithm.

System introduced for 2016 (D-A with global tie-breaking):

- Same, but use just a single lottery for all schools.

Original system not stable or strategyproof, while new systems are. Local tie-breaking fairer, but less efficient (more swaps desired).

Summary: Matching

We have seen several **variants** of the stable matching problem:

- basic marriage problem, extension to incomplete preferences, extension to preferences with ties
- we have hinted at possible extensions to many-to-one variants

We have discussed various desirable **properties** of matchings:

- stability: no couple has an incentive to break the matching
- efficiency: no mutually beneficial swaps possible after assignment
- strategyproofness for one side of the market: no incentive to lie
- strategyproofness for both sides: incompatible with stability
- fairness: possibly expressed in terms of "regret"
- maximality (in terms of cardinality): computationally intractable

We have seen how the deferred-acceptance algorithm of Gale and Shapley can be used to compute stable matchings efficiently.