Matching under Preferences

Marco Chiarandini
Lektor

Department of Mathematics & Computer Science
University of Southern Denmark

Optagelsessystemet (KOT)

Snart vil du ansøge for optagelsen i en vidergående uddannelse

Du kan søge op til 8 forskellige uddannelser i prioriteret rækkefølge (i gennemsnit søger man 2,8 uddannelser). Det kan både være kvote 1- og kvote 2-ansøgninger. Men du kan højst blive optaget på én uddannelse ét sted.

Mest uddannelsesteder har et begrænset antal pladser og giver preference til studerende med højere karaktergennemsnit.

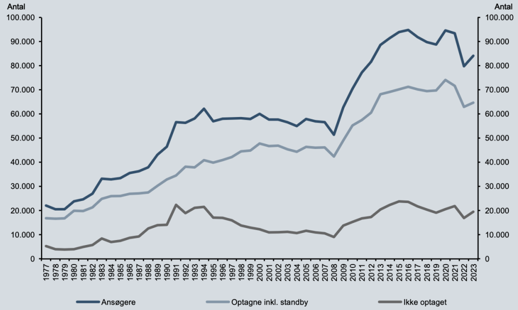
I 2020 var der 94.599 ansøgerer der søgte en af de ca. 900 uddannelser af disse ca. 75.000 (inkl standby) blev optaget.
Af disse 30.578 blev optaget i en af de 100 bacheloruddannelser.
(Kilde: Den Koordinerede Tilmeldings database og Danmarks Statistik).

Hvordan fungerer tildelingsproceduren?

KOT HOVEDTAL

2023

28. juli



Matching under preferences

Matching agents one to another

Examples:

- pupils to schools
- junior doctors to hospitals
- kidney patients to donors
- roommate assignments

subject to ordinal preferences over a subset of the others. That is, there is a ranking or list of preferences with first choice, second choice, etc., on both sides

Other issues:

- capacity constraints
- large scale applications: in Hungary in 2011, 140.953 students applied for admission at universities; In US National Resident Matching Program in 2012, 38.777 aspiring residents, 26.772 available positions.
- free-for-all markets: free negotiations: issues of unraveling, congestion, exploiting offers

Centralized Matching Schemes

Third party computes (automatically) optimal matching

pursuing one or more of these criteria:

- maximizing the number of places filled in the educational programmes,
- giving the maximum number of applicants their first-choice programme,
- ensuring no applicant and programmes have an incentive to reject their assignees and become matched together.

How can all this be done?

David Gale and Lloyd S. Shapley *College Admission and the Stability of Marriage* The American Mathematical Monthly, Vol. 69, No. 1. 1962

COLLEGE ADMISSIONS AND THE STABILITY OF MARRIAGE

D. GALE* AND L. S. SHAPLEY, Brown University and the RAND Corporation

1. Introduction. The problem with which we shall be concerned relates to the following typical situation: A college is considering a set of n applicants of which it can admit a quota of only q. Having evaluated their qualifications, the admissions office must decide which ones to admit. The procedure of offering admission only to the q best-qualified applicants will not generally be satisfactory, for it cannot be assumed that all who are offered admission will accept. Accordingly, in order for a college to receive q acceptances, it will generally have to offer to admit more than q applicants. The problem of determining how many and which ones to admit requires some rather involved guesswork. It may not be known (a) whether a given applicant has also applied elsewhere; if this is known it may not be known (b) how he ranks the colleges to which he has applied; even if this is known it will not be known (c) which of the other colleges will offer to admit him. A result of all this uncertainty is that colleges can expect only that the entering class will come reasonably close in numbers to the desired quota, and be reasonably close to the attainable optimum in quality.

The usual admissions procedure presents problems for the applicants as well as the colleges. An applicant who is asked to list in his application all other colleges applied for in order of preference may feel, perhaps not without reason, that by telling a college it is, say, his third choice he will be hurting his chances of being admitted.

One elaboration is the introduction of the "waiting list," whereby an applicant can be informed that he is not admitted but may be admitted later if a vacancy occurs. This introduces new problems. Suppose an applicant is accepted by one college and placed on the waiting list of another that he prefers. Should he play safe by accepting the first or take a chance that the second will admit him later? Is it ethical to accept the first without informing the second and then withdraw his acceptance if the second later admits him?

We contend that the difficulties here described can be avoided. We shall describe a procedure for assigning applicants to colleges which should be satisfactory to both groups, which removes all uncertainties and which, assuming there are enough applicants, assigns to each college precisely its quota.

2. The assignment criteria. A set of n applicants is to be assigned among m colleges, where q_i is the quota of the ith college. Each applicant ranks the colleges in the order of his preference, omitting only those colleges which he would never accept under any circumstances. For convenience we assume there are no ties; thus, if an applicant is indifferent between two or more colleges he is nevertheless required to list them in some order. Each college similarly ranks the students who have applied to it in order of preference, having first eliminated those appli-

* The work of the first author was supported in part by the Office of Naval Research under Task NR047-018.

Stable Matching – **Problem Statement**

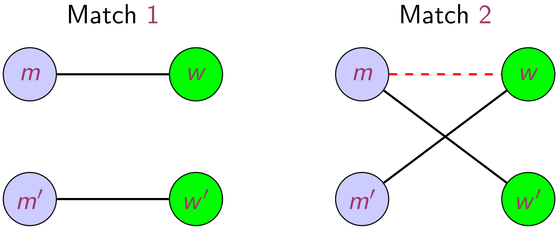
- A matching is collection of pairs of agents in such a way that no agent is in more than one pair
- A stable matching is a matching in which two agents cannot be found, who would prefer each other over their current counterparts (unstable pair).

Formalization:

Input:
n men and n women, where each person has ranked all members of the opposite sex with a unique number between 1 and n in order of preference.
Output:
A matching of the men and women with no unstable pair.

Two men and two women: unstable matching

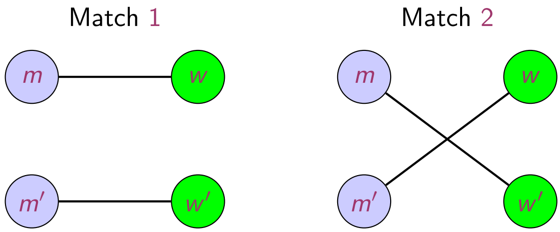
- Example 1: (consensus preference: 1 stable matching)
  m prefers w to w';
  m' prefers w to w';
  w prefers m to m';
  w' prefers m to m';



- In matching 2, m and w form an `unstable pair:` (red, dashed line)
  — both m and w prefer the other to their current partners;

Two men and two women: stable matching

- Example 2: (different preference: 2 stable matchings)
  m prefers w to w';
  m' prefers w' to w;
  w prefers m' to m;
  w' prefers m to m';



- Both matching 1 and 2 are stable.

Three men and three women: unstable matching

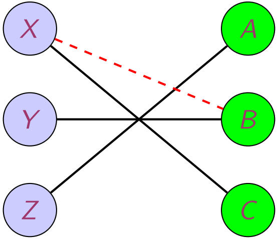
	favorite ↓ 1st	2nd	least favorite ↓ 3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

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

Women's Preference Profile

- Is matching $X - C,\ \ Y - B,\ \ Z - A$ stable?
- No. Bertha and Xavier will unravel and engage with each other.



Three men and three women: stable matching

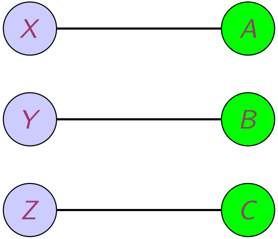
	1st (favorite)	2nd	3rd (least favorite)
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

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

- The matching $X - A,\ Y - B,\ Z - C$ is stable.



Questions

Does there always exist a stable marriage?
If yes, how can we always find it?
What is the computational cost?

Question

Does there always exist a stable marriage?
How to find a stable matching?

Constructive proof

Trial 0: Brute Force

Enumerate all possible matchings +
Stability Checking Algorithm

```
for m := 1 to n do
    for each w such that m prefers w to M(m) do
        if w prefers m to M(w) then
            return unstable;
    return stable;
```

How many matchings are there? How many operations overall?

I 2015 var der 92.477 ansøgerer, af disse 40.565 søgte en af de 100 bacheloruddannelser. (Kilde: Den Koordinerede Tilmeldings database)

$$40565! \approx 1.6304 \times 10^{169315}$$

A desktop computer (Intel Core i7 5960X) does 238310×10^6 instructions per second (MIPS)

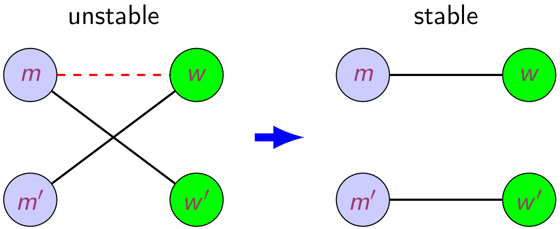
$$\frac{40565!}{238310 \times 10^6 \times 31536000} \approx 2224 \times 10^{169299} \text{ years}$$

Trial 1: Improvement strategy

Trial 1: Improvement strategy

- Basic idea: start from a **complete** matching, and try to **improve** the matching via reducing unstable pairs. If the number of unstable pairs is reduced to 0, then we get a solution.
- Switching operation: making unstable pairs to be stable
- An example of Switching operation:

  m prefers w to w';
  m' prefers w to w';
  w prefers m to m';
  w' prefers m to m';



Trial 1: Improvement strategy

Initializing a matching randomly;
while there exist unstable pairs **do**
  Select an unstable pair $m - w$ arbitrarily ;
  Perform Switching operation to resolve the unstable pair $m - w$;

Improvement strategy: a success case

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

favorite ↓ (1st), least favorite ↓ (3rd)

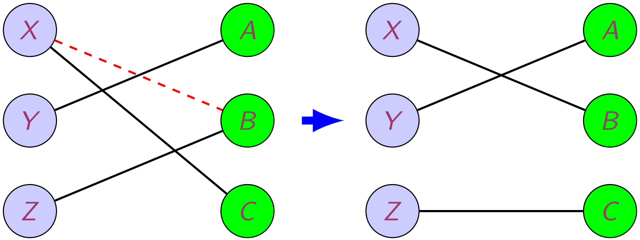
Men's Preference Profile

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

favorite ↓ (1st), least favorite ↓ (3rd)

Women's Preference Profile

- Starting from an unstable matching. After one step of switching, we get a stable matching.

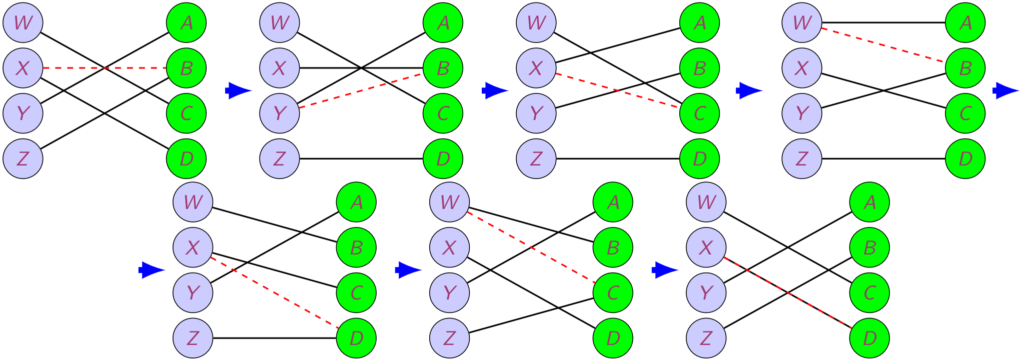


Improvement strategy: a failure case

	man			
	1st	2nd	3rd	4th
W	C	B	D	A
X	B	D	C	A
Y	C	D	B	A
Z	D	C	B	A

	woman			
	1st	2nd	3rd	4th
A	Z	W	Y	X
B	W	Y	X	Z
C	X	Y	W	Z
D	X	Y	W	Z

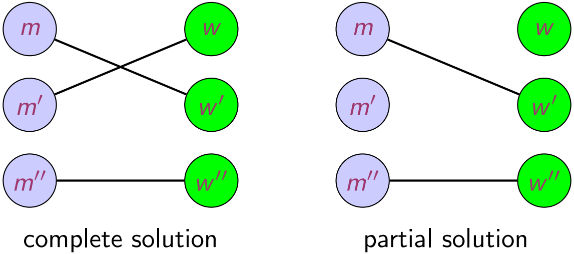
Starting from an unstable matching Step 1, 2, 3, 4, 5, 6. Failed! Return to the initial matching.



Trial 2: Construction strategy

Trial 2: Construction strategy

- Key observation: the solution is a **compelte** matching.
- Basic idea: Growing up from **partial** matching to **complete** matching, and ensure no unstable pairs during the increment process.
- Implementation: a "propose-engage" process.
  Man: propose, woman: accept or reject.



Stable Matching – Gale_Shapley algorithm

Day 1 Each man proposes to his choice n. 1.
Some women will receive some proposals, other none.
Each woman reject all suitors except the top ranked.
Tentative engagements are formed.

Day 2 Each rejected man proposes to his next best choice independently on whether she is engaged or not.
Each woman rejects all but her top suitors

Day 3,4,5,... Repeat as in Day 2

Stable Matching – Gale_Shapley algorithm

A bit more formal algorithm:

```
assign each person to be free
while some man m is free do
    w = first woman on m's list to whom m has not yet proposed
    if w is free then
        assign m and w to be engaged (to each other)
    else
        if w prefers m to her fiance' m' then
            assign m and w to be engaged and m' to be free
        else
            w rejects m (and m remains free)
```

Correctness proof

Key observations of Gale_Shapley algorithm

Key observations:

1. Men propose to women in the decreasing order of preference.
2. Once a woman is matched, she never becomes unmatched.
3. When a man proposes, the existing matching might be destroyed.

Correctness: perfection

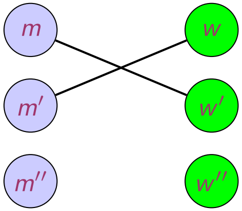
Theorem

All men and women finally get matched and the algorithm terminates.

Bevis.

Suppose m'' is not matched upon termination;

- then there is woman, say w'', is not matched;
- then w'' should be never proposed to (by Observation 2);
- But m'' proposes to everyone. Contradiction.



□

Correctness: stability

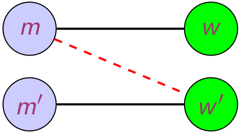
Theorem

At each step of the while loop, the intermediate partial match is a stable match. As a special case, the finally reported match S^* contains no unstable pairs.

Bevis.

Suppose $m - w'$ is an unstable pair: each prefers the other to the current partner in S^*;

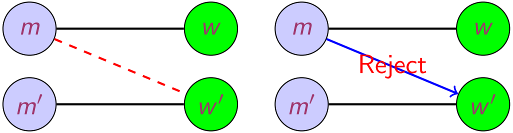
- Case 1: m never proposed to w'
  $\Rightarrow$ m prefers his GS partner w to w'
  $\Rightarrow$ $m - w'$ is stable. A contradiction.



□

Correctness: stability

- Case 2: m has proposed to w'
  $\Rightarrow$ m should be rejected by w'
  $\Rightarrow$ w' prefer her GS partner m' to m
  $\Rightarrow$ $m - w'$ is stable. A contradiction.



Corollary

A stable marriage always exists.

Algorithm analysis: time complexity and space complexity

Analysis: time-complexity

Theorem

Gale-Shapley algorithm ends in $O(n^2)$ steps.

Bevis.

- Key: find a measure of progress for this $\mathit{while}(1)$ type loop;
- Measure: the number of tried proposals $\#P$;

(see an extra slide)

□

Analysis: time-complexity

- Each step: $\#P$ increases at least 1;
- Upper bound: $\#P \leq n^2$
- So $T(n) = \#Step \leq n^2$;

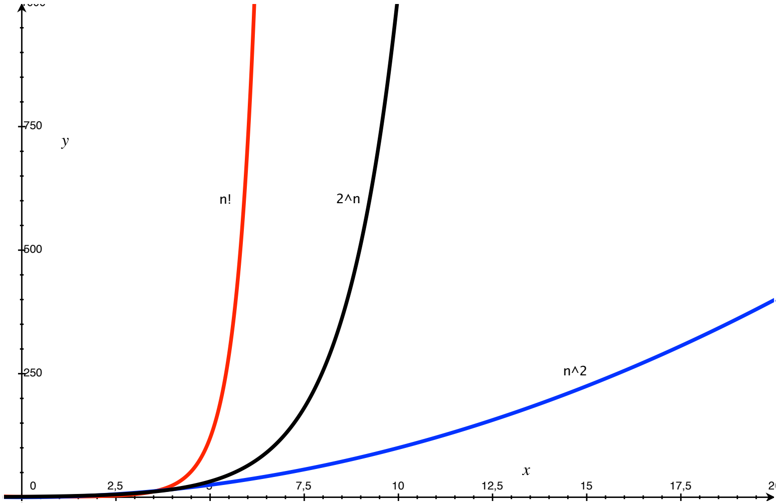
Time complexity and space complexity

- Time (space) complexity of an algorithm quantifies the time (space) taken by the algorithm.
- Since the time costed by an algorithm grows with the size of the input, it is traditional to describe running time as a function of the input size.
  - **input size**: The best notion of input size depends on the problem being studied.
    - For the Stable Matching problem, the `number of items in the input`, i.e. the number of men, is the natural measure.
    - For the Multiplication problem, the `total number of bits` needed to represent the input number is the best measure.

Running time: we are interested in its growth rate

- Several simplifications to ease analysis of Gale-Shapley algorithm:
  1. We simply use the number of primitive operations (rather than the exact seconds used) under the assumption that a primitive operation costs constant time. Thus the running time is $T(n) = an^2 + bn + c$ for some constants a, b, c.
  2. We consider only the leading term, i.e. an^2, since the lower order terms are relatively insignificant for large n.
  3. We also ignore the leading term's coefficient a since it is less significant than the growth rate.
- Thus, we have $T(n) = an^2 + bn + c = O(n^2)$. Here, the letter O denotes order.

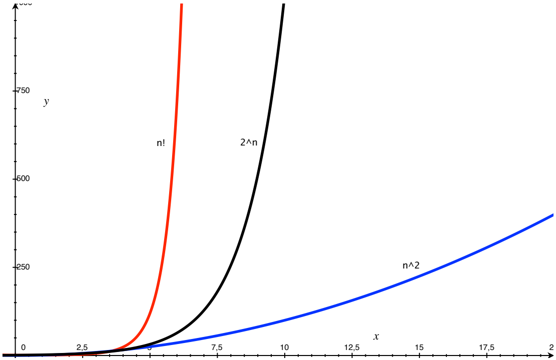
Polynomial vs Exponential and Factorial growth



Big O notation

- Recall that big O notation is used to describe the `error term` in Taylor series, say:

$$e^x = 1 + x + \frac{x^2}{2} + O(x^3) \text{ as } x \to 0$$



Example: $f(x) = O(g(x))$ as there exists $c > 0$ (e.g. $c = 1$) and $x_0 = 5$ such that $f(x) < cg(x)$ whenever $x > x_0$

Big Ω and Big Θ notations

- In 1976 D.E. Knuth published a paper to justify his use of the Ω-symbol to describe a stronger property. Knuth wrote: "For all the applications I have seen so far in computer science, a stronger requirement . . . is much more appropriate".
- He defined
$$f(x) = \Omega(g(x)) \Leftrightarrow g(x) = O(f(x))$$
with the comment: "Although I have changed Hardy and Littlewood's definition of Ω, I feel justified in doing so because their definition is by no means in wide use, and because there are other ways to say what they want to say in the comparatively rare cases when their definition applies".
- Big Θ notation is used to describe "$f(n)$ grows asymptotically as fast as $g(n)$".
$$f(x) = \Theta(g(x)) \Leftrightarrow g(x) = O(f(x)) \text{ and } f(x) = O(g(x)).$$

Space Complexity – Gale_Shapley algorithm

```
assign each person to be free
while some man m is free do
    w = first woman on m's list to whom m has not yet proposed
    if w is free then
        assign m and w to be engaged (to each other)
    else
        if w prefers m to her fiance' m′ then
            assign m and w to be engaged and m′ to be free
        else
            w rejects m (and m remains free)
```

Space Complexity – Gale_Shapley algorithm

Data structures

```
for m = 1 to M do
    partnerForMan[m] = NULL
for w = 1 to W do
    partnerForWoman[w] = NULL
while TRUE do
    if there is no man m such that partnerForMan[m] = NULL then
        return;
    select such a man m arbitrarily;
    w = the first woman on m's list to whom m have not yet proposed;
    if partnerForWoman[w] == NULL then
        partnerForWoman[w] = m; partnerForMan[m] = w;
    else if w prefers m to partnerForWoman[w] then
        partnerForMan[partnerForWoman[w]] = NULL;
        partnerForWoman[w] = m;
        partnerForMan[m] = w;
    else
        //do nothing means simply rejecting m;
```

Are there different ways to solve the problem?
How do they compare?
What would the best possible assignment for a man in a stable marriage?

Other Results

Theorem

Any execution of Gale_Shapley algorithm yields the same matching S^*.

Note:

- The theorem is non-trivial since in line 11, an unmatched man m is selected arbitrarily.

Notations:

- **Valid partner:** w is a valid partner of m if the pair $m - w$ exists in a stable match;
- **Man-optimal match:** each m pairs with his best valid partner, i.e., the best choice he can get.

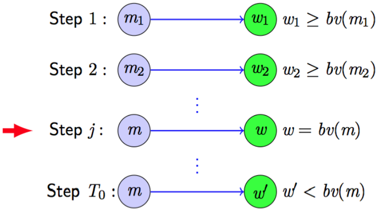
Lemma

The Gale_Shapley algorithm generates a **man-optimal match** S^*.

Notation: for a man m, we indicate by $w \succ_m w'$ if w is ranked higher than w' in the list of m.

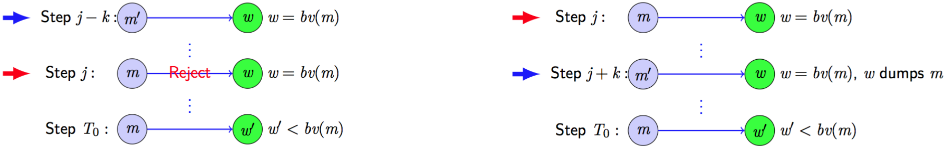
Proof

- A proposal is called "unlucky" if the man proposes to a woman with rank lower than his best valid partner ($w' \prec_m best\_valid(m)$).
- For the sake of contradiction, suppose there is at least one unlucky proposal in an execution.
- Let $T = \{t \mid$ at step t a man proposes to a woman with rank lower than his best valid partner $\}$. Let $T_0 = \min\ T$, i.e. the first unlucky proposal occurs.
- Suppose at time T_0 it is m that proposes to w' such that $w' \prec_m best\_valid(m)$. '
- Thus before step T_0, m should have proposed to his best valid partner (denoted as w) since w is ranked more highly than w'.



Proof cont'd

- But m finally didn't pair with w. Why?
  There are two cases:
  1. m was rejected by w directly: w has already paired with m' and in her rank list, m' is better than m (see left-hand panel).
  2. m was accepted by w but was dumped by w afterwards: m' is proposing w and in her rank list, m' is better than m (see right-hand panel).



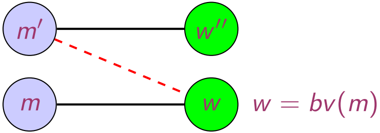
In both cases, the following property holds:

1. For w: w prefers m' to m.
2. For m': $w \succeq_{m'} best\_valid(m')$ since T_0 is the first time that an "unlucky" proposal occurs.

Proof cont'd

- The fact that w is best valid partner of m means that there exists a stable matching, denoted as S', where m pairs with w. Suppose that m' pairs with w'' in S'.

Stable match S'



- Then $m' - w$ should be an unstable pair. (Why?)
- A contradiction. In other words, unlucky proposals never occur in the "propose-engage" process, and any executions of the algorithm yields the same stable matching.

Corollary

The Gale_Shapley algorithm assings every woman to her worse valid man.

Summary

- Matching under preferences
- Gale Shapley (1962) algorithm for stable matchings.
- Algorithm design
- Algorithm analysis (correctness, properties, running time)
- Algorithm implementation
- Math and Computer Science: from the real world to abstractions and return

Context

In economics (game-theory):

- matching theory as part of market design in microeconomics
- matching algorithm as a mechanism
- interest in strategy-proof or truthful mechanisms: make a dominant strategy for the agents to reveal their true preferences

In CS

- Computational social choice (collective decisions) -> Algorithmic mechanism design (social welfare)
- Algorithmic Game Theory concerned with computational questions

Bibliography

- D. Gale and L.S. Shapley, College Admission and the Stability of Marriage. The American Mathematical Monthly, Vol. 69, No. 1. 1962
- D.E. Knuth. Stable Marriage and its Relation to Other Combinatorial Problems. American Mathematical Society, 1976 (translated to English 1997)
- D. Gusfield and R. Irving. The Stable Marriage Problem: Structure and Algorithms. MIT Press, 1989
- D.F. Manlove. Algorithmics of Matching Under Preferences. World Scientific, 2013.
- Videos on the Stable Marriage Problem by Numberphile, Featuring Dr Emily Riehl
  https://www.youtube.com/watch?v=Qcv1IqHWAzg
  https://www.youtube.com/watch?v=LtTV6rIxhdo
- http://optimalmatching.com/

Variants and Extensions

- Indifference in agents' lists, ie ties
- Incomplete/bounded lists
- Exchange stability: no pair of residents who could exchange one another's assigned hospitals so as to improve their outcome
- tripartite matching problem with preferences
- find all stable matchings
- find stable matching with other properties

Classification

- Bipartite matching problems with two-sided preferences
  - Stable Marriage problem (SM)
  - Hospitals Resident problem (HR) (many-one SM generalization)
    Workers Firm problem, Student-Project Allocation problem

  Optimality criteria: Stability: no two agents prefer another to one of their current assignees

- Bipartite matching problems with one-sided preferences
  - House Allocation problem (HA)
  - Capacited House Allocation Problem (CHA) (many-one HA generalization)

  Optimality criteria: Pareto optimality, popularity, profile-based optimality

- Non-bipartite matching problems with preferences
  - Stable Roommates problem (SR)
    chess players, kidney exchanges patient-donor, P2P network
  - Stable Fixtures, S. Multiple Activities, S. Allocation (many-many)
  - Coalition Formation Game (partnerships of size > 2)

  Optimality criteria: Stability

Matching under Preferences

Marco Chiarandini
Lektor

Department of Mathematics & Computer Science
University of Southern Denmark