26 Jan 2018

ADMINISTRATIVE

1. You sign up for cs4820 on piazza.com.
   Use this for receiving announcements from instructors, asking questions to instructors and/or peers.
   Questions can be private (instructor only) or public.
   If your question discloses how to solve a homework problem, make it private.
   If your question addresses a confusion that may affect other students, please make it public.

2. We sign you up for CMS, if you're enrolled in 4820.
   Write a private question to the instructors on Piazza if this hasn't happened.
   CMS is for you to download homework, turn it in, download lecture notes and homework solution sets, look at your grade and grader's comments, request regrades.

3. The first assignment is out on CMS.
   3 questions: two problem solving
   one coding
   Due 11:59pm Thurs, 1 Feb 2018.
   Grace period is > 0 minutes, < 30 minutes.
   You may work in groups of ≤ 4.
   Write up solutions separately and in your own words!

4. Academic integrity is paramount.
   Don't copy someone else's words or code.
   Don't use web sources to obtain HW solutions or code.
   If you accidentally stumble upon something that you use in a HW solution, acknowledge your source.

(5) Office hours begin Monday.
Schedule will be announced (Piazza, 4820 website)
over this weekend.

(6) "Slip days": 6 per student, for the whole semester.
To be used, in whole-number increments, when circumstances (e.g. temporary illness) delay completing an assignment.
Use your budget wisely!

Stable Matching (Section 1.1)

Often used for matching workers to job positions.

Input.
n applicants
n job positions ("employers")

(simplifying assumption: # applicants = # positions avail.)

Each applicant submits a ranking of employers.
" employer " " " " applicants.

Output. Perfect matching of applicants to employers.

i.e. a set of ordered pairs $(\text{applic}_i, \text{empl}_j)$

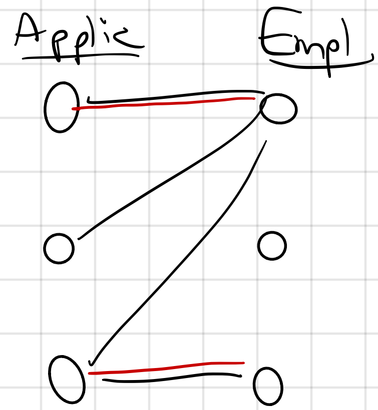
such that (a) no two pairs have a member in common (an applic. or empl.)

"matching"

(b) every applic. and every empl. belongs to one of the pairs

"perfect"

What makes a perfect matching "good"?

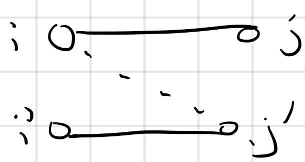


("Maximum matching", Chapter 7)

This formulation and its generalization to maximum weight matching (also Chapt. 7) have some issues with unfairness and/or dishonest reporting of preferences.

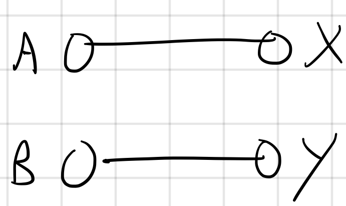
Gale & Shapley (1962) formulated "stability" as an objective.

In the perfect matching M that we output, there should not exist two pairs (i, j) and (i', j') such that

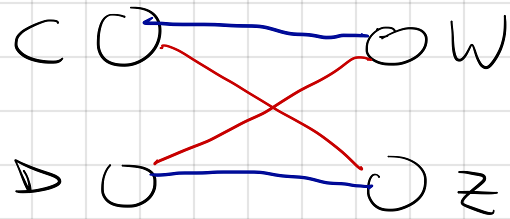


i prefers j' to j
j' prefers i to i'.

Examples.



Preferences. A: $X > Y$ B: $X > Y$
X: $A > B$ Y: $A > B$



C: W > Z   D: Z > W

W: D > C   Z: C > D

Red & blue matchings are both stable.

Does a stable perfect matching always exist?

Not obvious. The answer is yes. Gale & Shapley proved it by giving an algorithm that is guaranteed to compute a stable perfect matching.

The Proposal Algorithm

Start with everyone unmatched.

while there exists an unmatched employer, x:

  x "proposes" to the highest ranked applicant it has not yet proposed to, y.

  if y is matched and prefers its current match to x, make no change to the matching.

  else (x, y) become matched to each other. previous partner of y (if any) becomes unmatched.

endwhile

Remark: This algorithm, like many others, is underspecified. If more than one employer is unmatched at the start of the "while" block, the algorithm does not specify which one to choose. Any implementation of the algorithm that conforms to the specification (i.e. chooses an unmatched x whenever one exists) is considered to be a valid implementation.