Algorithmic Game Theory

Stable Matching

Joseph Chuang-Chieh Lin
Dept. CSIE, Tamkang University

Stable Matching

- A matching is said to be *unstable* if there exist 2 marriage couples *X*-*x* and *Y*-*y* such that *X* desires *y* more than *x* and *y* desires *X* more than *Y*.
-
- The pair *X*-*y* is said to be "dissatisfied." (不滿的)
-
- A marriage *M* is called "*stable marriage*" if there is no dissatisfied couple.

Example

- For example, $N = 4$.

A: *abcd* B: *bacd* C: *adcb* D: *dcab*

a: $ABCD$ *b*: $DCBA$ *c*: $ABCD$ *d*: $CDAB$

- Consider a marriage M: A-*a*, B-*b*, C-*c*, D-*d*,
- C-*d* is dissatisfied. Why?

The Proposal Algorithm

- <u>**Proposal algorithm**</u>:

  Assume that the schools are numbered in some arbitrary order.

- The **lowest** numbered unmatched school *X* proposes to the **most** desirable student on its list who has not already rejected him; call her *x*.

The Proposal Algorithm (contd.)

- The student *x* will accept the proposal if she is currently unmatched, or if her current matched school *Y* is less desirable to her than *X*.
- The algorithm repeats this process, terminating when every school is matched.
  - Such an algorithm is used by hospitals in North America in the match program that assigns medical graduates to residency positions.

Discussion

- Does it always terminate with a stable marriage?

Discussion

- Does it always terminate with a stable marriage?
- An unmatched school always has at least one student available that he can propose.

Discussion

- Does it always terminate with a stable marriage?
- An unmatched school always has at least one student available that he can propose.

- At each step the proposer will eliminate one student on his list and the total size of the lists is n^2.
  - Thus the algorithm uses at most n^2 proposals. i.e., it always terminates.

<u>**Claim**</u>: The Final Matching is Stable

- *Proof by contradiction:*

  Let *X*-*y* be a dissatisfied pair, where in *M* they are paired as *X*-*x*, *Y*-*y*.

  Since *X* prefers *y* to *x*, he **must have proposed to *y*** before getting matched to *x*.

  Since *y* either rejected *X* or accepted it only to abandon it later, her mates thereafter (including *Y*) must be more desirable to her than *X*.

  Therefore *y* must prefer *Y* to *X*, contradicting the assumption that *y* is dissatisfied.

Principle of Deferred Decisions

- The idea is to assume that the entire set of random choices is ***not*** made in advance.

- At each step of the process, we fix only ***the random choices*** that must be revealed to the algorithm.

Principle of Deferred Decisions

- Suppose that schools do not know their lists to start with. Each time a school has to make a proposal, he picks a random student from the set of students who has not already been proposed by him, and proceeds to propose to her.

- The only dependency that remains is that the random choice of a student at any step depends on the set of proposals made so far by the current proposer.

Amnesiac Algorithm

- However, we can eliminate the dependency by modifying the algorithm, i.e., a school chooses a student uniformly at random from the set of all n students, including those to whom it has already proposed to.

- It *forgets* the fact that these students have already rejected him.

- We call this new version the *Amnesiac Algorithm*.

Amnesiac Algorithm

- Note that a school making a proposal to a student who has already rejected it will be rejected again.
- Thus, the output by the Amnesiac Algorithm is exactly the same as that of the original Proposal Algorithm.
- The only difference is that there are some *wasted* proposals in the Amnesiac Algorithm.