CS 70 Spring 2024 — Discrete Mathematics and Probability Theory — Seshia, Sinclair — DIS 1B

# 1 Stable Matching

Note 4

Consider the set of jobs $J = \{1, 2, 3\}$ and the set of candidates $C = \{A, B, C\}$ with the following preferences.

| Jobs | Candidates |
|---|---|
| 1 | A > B > C |
| 2 | B > A > C |
| 3 | A > B > C |

| Candidates | Jobs |
|---|---|
| A | 2 > 1 > 3 |
| B | 1 > 3 > 2 |
| C | 1 > 2 > 3 |

Run the traditional propose-and-reject algorithm on this example. How many days does it take and what is the resulting pairing? (Show your work.)

| | D1 | D2 | D3 | D4 | D5 |
|---|---|---|---|---|---|
| A | 1,3 | 1 | 1,2 | 2 | 2 |
| B | 2 | 2,1 | 3 | 1,3 | 1 |
| ~~C~~ | | | | | 3 |

Result: (A,2), (B,1), (C,3)

# 2 Propose-and-Reject Proofs

Note 4

Prove the following statements about the traditional propose-and-reject algorithm.

(a) In any execution of the algorithm, if a candidate receives a proposal on day $i$, then she receives some proposal on every day thereafter until termination.

obvious

(b) In any execution of the algorithm, if a candidate receives no proposal on day $i$, then she receives no proposal on any previous day $j$, $1 \le j < i$.

use (a)

(c) In any execution of the algorithm, there is at least one candidate who only receives a single proposal. (Hint: use the parts above!)

Suppose finish date is k, on k-1, at least one candidate, say $m_1$, didn't receive any proposal. If this was not true, then by (a), everyone has at least one proposal on k-1 and only one job position, as by (a). But then k-1 would be execution finish date. contradiction, thus $m_1$ mustn't have proposal. Thus on k, $m_1$ had only one proposal.

## 3 Be a Judge

Note 4

By stable matching instance, we mean a set of jobs and candidates and their preference lists. For each of the following statements, indicate whether the statement is True or False and justify your answer with a short 2-3 line explanation:

✓ output of algo

(a) There is a stable matching instance for $n$ jobs and $n$ candidates for $n > 1$, such that in a stable matching algorithm with jobs proposing, every job ends up with its least preferred candidate.

False. as for jobs to be paired up with least preferred candidate, needs to be rejected n-1. While at least one candidate only had one proposal. Thus contradiction.

(b) In a stable matching instance, if job $J$ and candidate $C$ each put each other at the top of their respective preference lists, then $J$ must be paired with $C$ in every stable pairing.

True. If exist stable matching $(C, J^*)$, $(C^*, J)$ obviously they are rogue couple. Thus contradiction. So must be true.

(c) In a stable matching instance with at least two jobs and two candidates, if job $J$ and candidate $C$ each put each other at the bottom of their respective preference lists, then $J$ cannot be paired with $C$ in any stable pairing.

False. eg.

| J | | C | |
|---|---|---|---|
| A | 1 2 | 1 | A B |
| B | 1 2 | 2 | A B |

$\Rightarrow$ (1, A), (2, B) $\rightarrow$ stable matching pair.

(d) For every $n > 1$, there is a stable matching instance for $n$ jobs and $n$ candidates which has an **unstable** pairing where **every** unmatched job-candidate pair is a rogue couple or pairing.

If exist matching such that each pair candidate and job are mutually least favorite Then, every pair is rogue pair.

## 4 Stable Matching III

Note 4

(a) True or False?

(i) If a candidate accidentally rejects a job she prefers on a given day, then the algorithm still always ends with a matching.

False eg

| | | |
|---|---|---|
| A | 1 | 2 |
| B | 2 | 1 |

| | | |
|---|---|---|
| 1 | A | B |
| 2 | B | A |

D1

1 rejects A

A does not offer 1 again while (B, 2) stable pair.

(ii) The Propose-and-Reject Algorithm never produces a candidate-optimal matching.

False Though Propose-reject Algo job optimal. But if every job's favorite candidate also favors the job back, then every candidate would match up with favorite job by 3(b). Thus the matching would be candidate optimal as well.

(iii) If the same job is last on the preference list of every candidate, the job must end up with its least preferred candidate.

False eg

| | | |
|---|---|---|
| A | 1 | 2 |
| B | 2 | 1 |

| | | |
|---|---|---|
| 1 | B | A |
| 2 | B | A |

match result would be (A, 1)(B, 2) with A not pairing with 2, the least favorite employee.

(b) As you've seen from lecture, the jobs-proposing Propose-and-Reject Algorithm produces an employer-optimal stable matching. Let's see if the candidate have any way of improving their standing. Suppose exactly one of the candidates has the power to arbitrarily reject one proposal, regardless of which job she has on her string (if any). Construct an example that illustrates the following: for any $n \geq 2$, there exists a stable matching instance for which using this power helps **every** candidate, i.e. every candidate gets a better job than she would have gotten under the jobs-proposing Propose-and-Reject Algorithm.

eg. A 1 2 | 1 B A

B 2 1 | 2 A B

D1 | 1 rejects A | D2 | D3
---|---|---|---
1: A | $\Rightarrow$ | | $\Rightarrow$ B
2: B | | B A | A