CS 70

Discrete Mathematics and Probability Theory

Spring 2024

Seshia, Sinclair

DIS 1B

1 Stable Matching

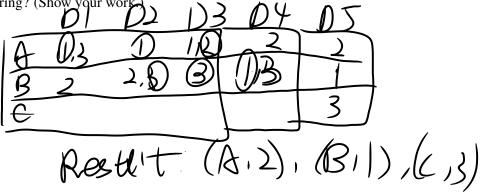
Note 4

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	>	В	>	С
2	В	>	A	>	С
3	A	>	В	>	С

Candidates	Jobs				
A	2 > 1 > 3				
В	1 > 3 > 2				
С	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)

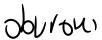


2 Propose-and-Reject Proofs

Note 4

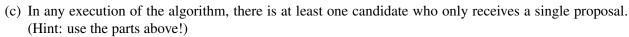
Prove the following statements about the traditional propose-and-reject algorithm.

(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.



(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)



Suppose tinkh date is k, on k-1, at least one candrate, say m, , drant receive any proposal It the was not true. Hen by ca), everyone has at least one proposal on k-1 and only one job position, as by cal But then R-1 would be execution tinkh date

By stable matching instance, we mean a set of jobs and candidates and their preference lists. For each of the

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:

(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 John to be parted up with least preferred candidate, needs to be rejected h-1 while at best one candidate only had one proposal thus contradictors

(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.

Three It exist stable menterny (c, J*), (b, J)
downously they are royue couple Thrus contraditions
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.

Note 4

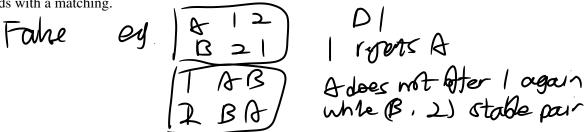
(d) For every n > 1, there is a stable matching instance for n jobs and n candidates which has an **unstable** pairing where every unmatched job-candidate pair is a rogue couple or pairing.

If exist matching such that each paint candidate and jobs are mutually least taxonte Then, every par 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.



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

False Though Propose-reject Algo job optimal
But if every jobs towar be candidate also twose
the Gob back, then every candidate would match up
with talor to job by 3(b) Thus the matching would

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

(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 $n \ge 2$, 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.

