Stable Matching

$J : \{j_1, j_2, \dots, j_n\}$ $C : \{c_1, c_2, \dots, c_n\}$

Rogue Couple: Some pair (j_x, c_y) that prefer each other over their matched partner.

Propose & Reject

Morning: Job proposes to top candidate on their list who hasn't rejected them.

Midday: Each candidate rejects all offers they recieve, save for their most preferred (which they "leave on a string")

Night: All jobs rejected cross off the candidate they proposed to in the morning (same candidate who rejected them during the day).

Stable: No rogue couples in output matching of PAR.

Improvement Lemma: Each candidate's job on string only gets more and more preferred.

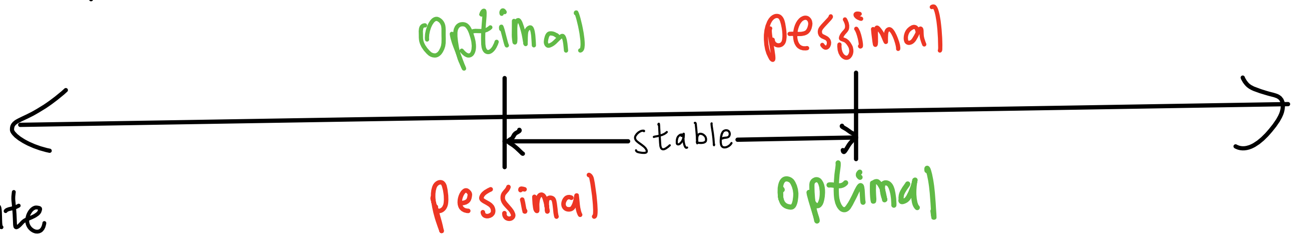
Proofs Advice: Contradiction, Induction, WOP

any nonempty subset of $\mathbb{N}$ has a smallest element.

Optimality & Pessimality

J_x : $C_1 > \cdots > C_i > \cdots > C_j > \cdots > C_n$

Job



Candidate

C_y : $J_n < \cdots < J_j < \cdots < J_i < \cdots < J_1$

CS 70 Discrete Mathematics and Probability Theory
Spring 2022 Koushik Sen and Satish Rao **DIS 1B**

1 Stable Matching

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

$\{(A,2), (B,1), (C,3)\}$

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.)

A	1, ~~3~~	1	~~1~~, 2	2	2
B	2	~~2~~, 3	3	~~3~~, 1	1
C					3

2 Propose-and-Reject Proofs

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

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

IH: day i+k, C receives an offer

IS: day i+k+1: C receive a offer

- receive an offer from J again ✓
- don't receive an offer from J ⇒ rejected J on day i+k, liked J* more than J

→ J* was left on string in day i+k

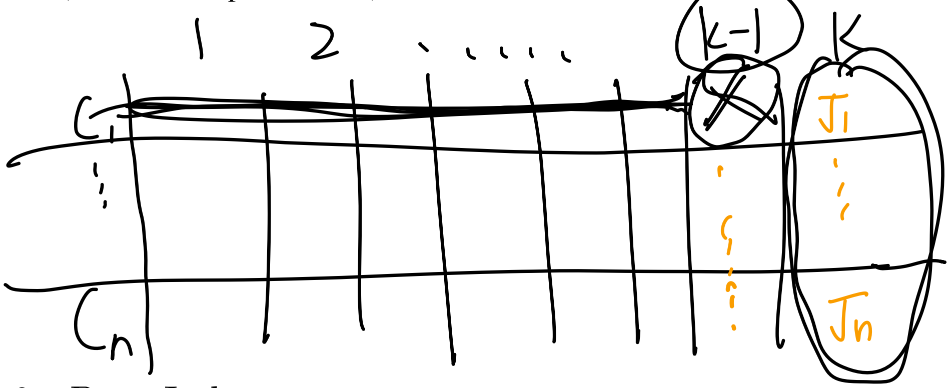
→ J* proposes to C on i+k+1 ✓

(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, $1 \le j < i$.

receive a job offer on any prev. day j ⇒ receive a proposal on day i > j

(contraposition)

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



C_1 received only 1 proposal overall

3 Be a Judge

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:

False

(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.

J_1 | $\overbrace{\cancel{\cdots}}^{n-1} > C_1$

J_2 | $\overbrace{\cancel{\cdots}}^{n-1} > C_2$

⋮

J_n | $\overbrace{\cancel{\cdots}}^{n-1} > C_n$

(J_1, C_1)

(J_2, C_2)

⋮

(J_n, C_n)

$n(n-1)$ total rejections

⇒ each candidate rejected $n-1$ jobs, Not possible.

True

(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.

Assume for sake of contradiction:

(J, C^*) is stable

(J', C)

⋮

$J: C > C^*$

$C: J > J'$

(C, J) is a rogue couple!

→ contradiction

(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

$J: \ldots > C$ $\quad$ $C: \ldots > J$

1	A	B
2	A	B

A	1	2
B	1	2

$(1, A)$ $\quad$ $(2, B)$

instance: set of n jobs & n candidates
each w/ preference lists

(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.

True

(J_1, C_1)
(J_2, C_2)
$\vdots$
(J_n, C_n)

$J_1 \mid \ldots > C_1$
$J_2 \mid \ldots > C_2$
$\vdots$
$J_n \mid \ldots > C_n$

$C_1 \mid \ldots > J_1$
$C_2 \mid \ldots > J_2$
$\vdots$
$C_n \mid \ldots > J_n$