CS 70 离散数学 和 概率论
DIS 1B
春季 2026 Sinclair, Song
Stable 匹配 Intro
Goal: 我们有 n candidates 和 n jobs; each candidate 和 job has a
preference list. We want to
Note4
pairupcandidatesandjobssuchthatthereisastablematching.
Definitions:
• StableMatchingInstance:
Asetofjobsandcandidatesandtheirrespectivepreferencelists
• 匹配: Disjointsetof(c, j)pairsthatarematchedtogether
i i
• Rogue Couple: a pair (c, j) 非 in the 匹配 that prefers each
other over their current
matchings(c, j′)和(c′, j)
• StableMatching: matchingwithnoroguecouples
• Optimalcandidateforjob:
highestrankedcandidateforajobinanystablematching
• Optimaljobforcandidate:
highestrankedjobforacandidateinanystablematching
• Job optimal: All jobs get their optimal candidates; a
candidate optimal 匹配 is defined
similarly
• Candidate pessimal: All candidates get their pessimal jobs
(i.e. lowest ranked job 对于 each

candidateinanystablematching);ajobpessimalmatchingisdefinedsimilarly
Proposeandrejectalgorithm: Eachday,
1. Morning: Jobs propose to the top candidate who have 非
rejected them; 笔记 that a job will

proposetothesamecandidateasthepreviousdayiftheywerenotrejected
2. Afternoon: Candidates say “maybe” to the best job offer
所以 far, keeping them in hand 或 on
astring,andsay“no”toeveryotheroffer
3. Evening: Jobscrossoffthecandidatesthathaverejectedthem
The process halts when no rejections happen; all candidates
那么 accept their current offer. 笔记
that the 算法 only produces one stable 匹配, 所以 there can be
other stable matchings 非
producedbythealgorithm.
Improvement 引理: every candidate will only say maybe to
better job offers as time goes on.

Similarly,everyjobwillonlyproposetoworsecandidatesastimegoeson.

Asaresult,theproposeandrejectalgorithmalwaysproducesastablematchingthatisjoboptimal
andcandidatepessimal.

Remember, a stable 匹配 instance is only defined as the 集合 of
jobs, candidates, 和
preferencelists.
Itdoesnotincludethematchingitself,noranyalgorithm.

1 Stable 匹配
考虑 the 集合 of jobs J ={1,2,3} 和 the 集合 of candidatesC
={A,B,C} with the following
Note4
preferences.

Jobs Candidates Candidates Jobs
1 A > B > C A 2 > 1 > 3
2 B > A > C B 1 > 3 > 2
3 A > B > C C 1 > 2 > 3
Runthetraditionalpropose-和-rejectalgorithmonthisexample.
Howmanydaysdoesittakeand
whatistheresultingpairing? (Showyourwork.)
解答:
The algorithmtakes 5days toproduce a 匹配. The
resultingpairing isas follows. Thecircles

indicatethejobthatacandidatepickedonagivenday(andrejectedtherest).
{(A,2),(B,1),(C,3)}.
Candidate Day1 Day2 Day3 Day4 Day5
A 1 ,3 1 1, 2 2 2
B 2 2, 3 3 1 ,3 1
C 3
2 Propose-和-Reject Proofs

Provethefollowingstatementsaboutthetraditionalpropose-和-rejectalgorithm.
Note4
(a) In any execution of the 算法, 如果 a candidate receives a
proposal on day i, 那么 they
receivesomeproposaloneverydaythereafteruntiltermination.
(b) In any execution of the 算法, 如果 a candidate receives no
proposal on day i, 那么 they
receivenoproposalonanypreviousday j,1≤ j<i.
(c) In any execution of the 算法, 存在 at least one candidate
who only receives a single
proposal. (提示: usethepartsabove!)
(d) There does 非 exist a stable 匹配 instance 对于 n jobs 和 n
candidates 对于 n>1, such
that in a stable 匹配 算法 with jobs proposing, every job ends
up with its least
preferredcandidate.
解答:
(a) Theideaistoinductonthenumberofdayspassedsofar.
Basecase: CandidateC receivesaproposalondayi
InductiveStep: AssumeCreceivesaproposalonday j≥ifromjobJ.
Wewanttoshowthey
CS70,Spring2026,DIS1B 3

will also get a proposal on day j+1. 存在 two cases: C prefers
J to all other offers, 或
C prefers some job J′ to J. In the first case J proposes toC
on day j+1 和 in the second J′
proposestoC onday j+1soC receivesatleastoneproposalonday
j+1.
(b) Onewayistouseaproofbycontradiction.
Assumethatacandidatereceivesnoproposalon
dayibutdidreceiveaproposalonsomepreviousday j,1≤ j<i.
Bythepreviouspart,因为
the candidate received a proposal on day j, they must
receive at least one proposal on every
dayafter j. Buti>
j,sothecandidatemusthavereceivedaproposalondayi,contradicting
ouroriginalassumptionthattheydidnot.
(c) 令’s say the 算法 takes k days. This means that every
candidate must have received
a proposal on day k. 然而, this also means that 存在 at least
one candidate C who

does 非 receive a proposal on day k 1 - 如果 this were 非 the
case, the 算法 would have
already terminated on day k 1. 那么 from part (b), sinceC did
非 receive a proposal on
day k 1, they didn’t receive a proposal on any day before k.
此外, we know they

gotexactlyoneproposalondayk,sincethealgorithmterminatedonthatday.
因此,wehave
thatC
receivesexactlyoneproposalthroughouttheentirerunofthealgorithm.
(d) 如果 such an instance existed, it would mean that at the
end of the 算法, every job would
have proposed to every candidate on its list 和 has been
rejected n 1 times. This also
means that every candidate got proposed to by every single
one of the jobs, 和 at the very
end must have rejected n 1 jobs in total (to end up with
only one preferred job at the last
day). We know this is impossible though, as we learned above
that at least one candidate
receivesasingleproposal.
因此,theremustbeatleastonecandidatewhoisnotproposedto
untiltheverylastday.
3 Be a Judge
对于 each of the following statements, indicate whether the
statement is 真 或 假 和 justify
Note4
youranswerwithashort2-3lineexplanation:
(a) In a stable 匹配 instance, 如果 job J 和 candidateC each put
each other at the top of their
respectivepreferencelists,thenJ mustbepairedwithC
ineverystablepairing.
(b) In a stable 匹配 instance with at least two jobs 和 two
candidates, 如果 job J 和 candi-
dateC
eachputeachotheratthebottomoftheirrespectivepreferencelists,thenJ
cannotbe
pairedwithC inanystablepairing.
(c)
Foreveryn>1,thereisastablematchinginstancefornjobsandncandidateswhichhasan

unstablepairingwhereeveryunmatchedjob-candidatepairisaroguecouple.
解答:
(a) 真: We give a simple 证明 by 矛盾. 假设 that J andC put each
other at the
top of their respective preference lists, 但是 J 和 C are 非
paired with each other in some
stable pairing S. 因此, S includes the pairings (J,C′),
(J′ ,C), 对于 some job J′ 和 candidate
C′ . 然而, J prefers C over its partner in S, 因为 C is at the
top of J’s preference list.

SimilarlyC prefers J over her current job. 因此 (J,C) form a
rogue couple in S, 所以 S is 非
stable. WehavearrivedatacontradictionthatS existswhereC andJ
arenotpaired.
因此 如果 job J 和 candidate C put each other at the top of their
respective preference

lists,thenJ mustbepairedwithC ineverystablepairing.
(b) 假: Thekeyhereistorealizethatthisispossibleifjob J
andcandidateCareatthebottom
of everybody else’s preference list as well. 对于 例子, a two
job, two candidate instance

wherethefirstjobisbestforbothcandidates,andthefirstcandidateisbestforbothjobshas
its only stable pairing 其中 the first job 和 first candidate
are paired (by part (b)), 和 the
second job 和 second candidate are paired. The second job 和
second candidate are each
other’sleastpreferredoption.
(c) 真: The key idea to this 解答 is that we want a 集合 of
preferences 对于 其中 J 和
i
C like each other the least 和 make a pairing J 和 C together.
In this 匹配 M each
i i i
unmatchedcoupleisaroguecouple.
Inparticular,aninstancecanbeconstructedbyforming
preferences 其中 job i’s (candidate i’s) least favorite
candidate is candidate i (job i) 和 an
arbitrary ordering on all the others. 因此, 对于 any job i 和
candidate j, 其中 i = j, 那么
job i prefers candidate j to i (its partner in M) 和
candidate j prefers job i to j (its partner
inM.) That,iseverypairiand j isaroguecouple.
Anexamplefortwojobsandtwocandidates,istheinstance:
Jobs Candidates
1 B A A 2 1
2 A B B 1 2

ThepairingP={(A,1),(B,2)}isthepairingwhereeachpairofjobsandcandidatesthatare
notinP,(A,2)和(B,1),areroguecouples.
4 Stable 匹配 III
(a) TrueorFalse?
Note4
(i) 如果 a candidate accidentally rejects a job they prefer on
a 给定 day, 那么 the 算法
stillalwaysendswithamatching.
(ii)
ThePropose-和-RejectAlgorithmneverproducesacandidate-optimalmatching.
(iii)
Ifthesamejobislastonthepreferencelistofeverycandidate,thejobmustendupwith
itsleastpreferredcandidate.
(b) As you’ve seen from lecture, the jobs-proposing
Propose-和-Reject 算法 produces an
employer-optimal stable 匹配. 令’s see 如果 the candidate have
any way of improving
their standing. 假设 exactly one of the candidates has the
power to arbitrarily reject one
proposal, regardless of 其中 job they have on their string (如果
any). Construct an 例子
thatillustratesthefollowing:
foranyn≥2,thereexistsastablematchinginstanceforwhich

usingthispowerhelpseverycandidate,即everycandidategetsabetterjobthantheywould
havegottenunderthejobs-proposingPropose-和-RejectAlgorithm.

解答:
(a) (i) 假,considerthecase:
Jobs Candidates
1 A B A 1 2
2 B A B 2 1
Using SMA, the 匹配 will be: (A,1),(B,2). 如果 candidate A
rejects job 1 despite
having no other jobs on their string, job 1 will propose to
candidate B 和 also get
rejected. This leaves both candidate A 和 job 1 partnerless.
In this case, the accidental
rejectionpreventsamatchingfrombeingproducedatall.
(ii) 假. 假设 that all jobs have a different first choice. Also
supposed that the job
proposing is each candidate’s first choice. In this case,
the 算法 would end after
the first day with both jobs 和 candidates ending with their
top pick. In this case, the
resultiscandidate-optimal.
(iii)
假,considerthefollowingcasewherejobsarenumbersandcandidatesareletters:
Jobs Candidates
1 A B A 2 1
2 B A B 2 1
Job 1 is last on every candidate’s list, 然而,{(A,1),(B,2)} is
a stable pairing 其中
job1gotitstopchoicecandidate.
(b) 不失一般性, 假设 that candidate 1 is the candidate with this
special power.
Now,assumethepreferencelistsareorderedasfollows:
Job Preferences Candidate Preferences
1 1>2>・・・>n 1>n 1 n>n 1>・・・>2>1
2 2>3>・・・>n 1>n>1 2 1>n>n 1>・・・>2
3 3>4>・・・>n>1>2 3 2>1>n>n 1>・・・>3
・・・ ・・・
n n>1>2>・・・>n 1 n n 1>n 2>・・・>1>n

IftheProposе-和-RejectAlgorithmwasrunwiththesepreferencelists,theneachcandidate
would be stuck with their least-preferred job. 然而, 令’s say
candidate 1 rejects job 1
on the first day, 即使 they have nobody on their string. 那么,
job 1 will be forced to
propose to its second option, candidate 2, 和 they will
accept because the job is their first
choice. Now, job 2 has no partner 和 will propose to
candidate 3, who will accept, leaving
job 3 without a partner. This process continues 直到 job n
proposes to candidate 1. One

rejectionhasledallcandidatestogettheirbestchoiceinsteadoftheirworstchoice!