EE 360C - ALGORITHMS

Lecture 4
Stable Matching

Vallath Nandakumar
University of Texas at Austin

Lloyd Shapley   Alvin Roth

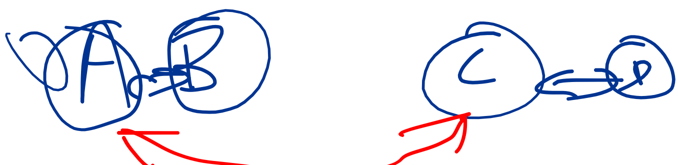
Agenda

- ☞ **Last week**
  - **Review of discrete math**
  - **Review of proof techniques**
    - **Contradiction, induction etc.**
- ☞ **Today**
  - **Stable Marriage problem**
- ☞ **Next class**
  - **Algorithm complexity**

INFORMAL PROBLEM

☞ Consider problem of optimally matching set of applicants to set of open positions

- Applicants to summer internships
- Applicants to graduate school
- Medical school graduate applications to residency programs
- Eligible males wanting to marry eligible females



STABILITY AND INSTABILITY

☞ Given set of preferences among hospitals and medical students, define a self-reinforcing admission process

☞ Unstable Pair

- Applicant *x* and Hospital *y* "unstable" if
  - *x* prefers *y* to its assigned hospital
  - *y* prefers *x* to one of its admitted students

☞ Stable Assignment

- No unstable pairs
  - Natural and desirable condition
  - Individual self-interest will prevent any applicant/ hospital deal being made

FORMULATING PROBLEM

- ☞ Consider set $M = \{m_1, \ldots, m_n\}$ of n men and set $W = \{w_1, \ldots, w_n\}$ of n women
  - Matching S - set of ordered pairs from $M \times W$ such that each member of M and each member of W appears in at most one pair in S
  - Perfect Matching S' – matching such that each member of M and each member of W appears in exactly one pair in S'
  - Each man $m \in M$ ranks all women
    - Referred to as "preference list"
  - Each woman ranks all men

FORMULATING PROBLEM

☞ Instability when matching *S* contains

- Two pairs *(m,w)* and *(m',w')* such that *m* prefers *w'* to *w* and *w'* prefers *m* to *m'*

☞ Goal: Perfect set of marriages with no instabilities

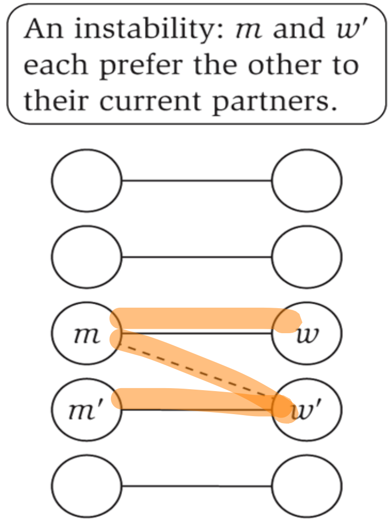


Figure 1.1 Perfect matching S with instability (m, w').

EXAMPLE

☞ **Is assignment *X-C, Y-B, Z-A* Stable?**

favorite ↓ … least favorite ↓

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite ↓ … least favorite ↓

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

QUESTIONS ABOUT STABLE MARRIAGE

☞ Does Stable Marriage exist for every set of preference lists?

☞ Given set of preference lists, can stable matching be efficiently constructed (if one exists)?

iClicker 1

Xavier prefers *Amy* to *Bertha*

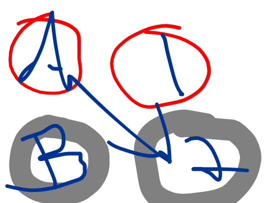
Yancey prefers *Amy* to *Bertha*

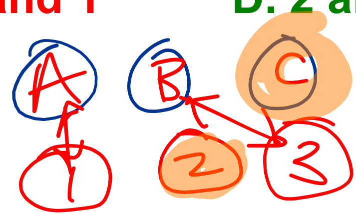
Amy prefers *Xavier* to *Yancey*

Bertha prefers *Xavier* to *Yancey*

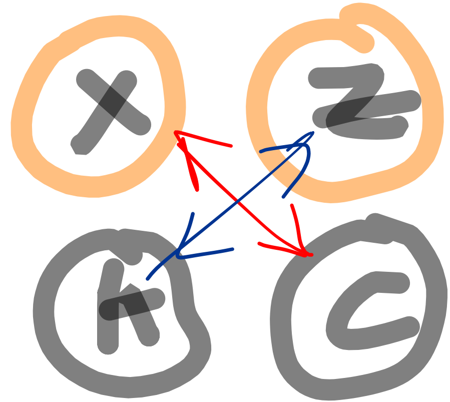
How many perfect matchings are possible? How many stable perfect matchings are possible?

A. 1 and 2 **B. 1 and 1** **C. 2 and 1** **D. 2 and 2**





iClicker 2



Zeus prefers *Amy* to *Clare*

Xavier prefers *Clare* to *Amy*

Amy prefers *Xavier* to *Zeus*

Clare prefers *Zeus* to *Xavier*

How many perfect matchings are possible? How many stable perfect matchings are possible?

A. 1 and 2 **B. 1 and 1** **C. 2 and 1** **D. 2 and 2**

GALE-SHAPLEY ALGORITHM <u>Demo</u>

```
Initially all m ∈ M and w ∈ W are free
while ∃m who is free and hasn't proposed to every w ∈ W
    do Choose such a man m
        Let w be the highest ranked in m's preference list
                to whom m has not yet proposed
        if w is free
            then (m,w) become engaged
            else w is currently engaged to m'
                if w prefers m' to m
                    then m remains free
                    else w prefers m to m'
                        (m,w) become engaged
                        m' becomes free
return the set S of engaged pairs
```

DOES IT WORK?

☞ Some Axioms

- *w* remains engaged from point when she receives first proposal
- Sequence of partners that *w* gets engaged to gets increasingly better (in terms of her preference list)
- Sequence of women to whom *m* proposes gets increasingly worse (in terms of his preference list)

☞ Observations

- Men propose to women in decreasing order of preference
- Once woman matched, she never becomes unmatched

TERMINATION

- What is good measure of progress of the algorithm?
  - Number of free men?
  - Number of engaged couples?
  - Number of proposals made?
- Each iteration consists of one man proposing to woman he has not proposed to before.
  - Count number of proposals
  - Iteration of while loop increases proposals by 1
  - Total number of proposals upper-bounded by n^2
- Theorem

n men each proposing n times

  - G-S algorithm terminates after at most n^2 iterations of while loop

TOWARDS PROVING PERFECT MATCHING

☞ Theorem

- If m is free at some point in execution of algorithm, then there is a woman he has not yet proposed to

☞ Proof

- Assume not true, m is free but has already proposed to every woman
  - Every woman must be engaged, otherwise would have said yes to m
  - If all n woman engaged, must be n engaged men
  - Contradicts claim m is free

PROVING PERFECT MATCHING

☞ Theorem

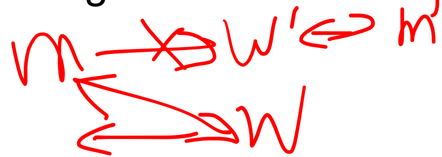
- Set S returned at termination is a perfect matching

☞ Proof

- Suppose algorithm terminates with free man m
  - Then m proposed to every woman (otherwise `while` loop still active and no termination)
  - Contradicts previous theorem that there cannot be a free man that has proposed to every woman

PROVING CORRECTNESS

- Theorem
  - The set of pairs *S* returned by G-S Algorithm is a stable matching



- Proof
  - Assume instability exists
    - There exists *(m,w)* and *(m',w')* such that *m* prefers *w'* to *w* and *w'* prefers *m* to *m'*
  - *m*'s last proposal must have been to *w*
    - Since *m* prefers *w'*, *m* must have proposed earlier to *w'*
      - *w'* must have rejected *m* in favor of *m''*
  - Either *m''* = *m'* or *w'* prefers *m'* to *m''*
    - Both contradict assumption that *w'* prefers *m* to *m'*

IMPLEMENTATION

- $O(n^2)$ implementation possible
- Representing Men and Women
  - Assume men named 1…*n*
  - Assume women named 1'…*n'*
- Engagements
  - Maintain list of free men in queue
  - Maintain two arrays of length *n*
    - *wife[m]* and *husband[w]*
    - Set entry to 0 if unmatched
    - If *m* matched to *w*, then *wife[m]=w* and *husband[w]=m*

IMPLEMENTATION

- ☞ Proposals
  - For each man, maintain list of women, ordered by preference
  - Maintain array *count[m]* that counts number of proposals made by man *m*
- ☞ Women Rejecting/Accepting
  - For each woman, create inverse of preference list
    - woman prefers *m* to *m'* if *inverse[m]* < *inverse[m']*

IMPLEMENTATION

☞ Proposal Process

- First free man *m* in queue proposes to woman at front of his preference list, *w*
- Increment *count[m]* and remove *w* from preference list
- *w* accepts proposal if unengaged or prefers *m* to her current match
- If *w* accepts, her former match goes back to queue of free men, otherwise *m* proposes to his next favorite

UNDERSTANDING SOLUTION

- For given problem instance
  - May be several stable matchings

- Example of instance with two stable matchings
  - A-X, B-Y, C-Z
  - A-Y, B-X, C-Z

	1st	2nd	3rd
Xavier	A	B	C
Yancey	B	A	C
Zeus	A	B	C

	1st	2nd	3rd
Amy	Y	X	Z
Bertha	X	Y	Z
Clare	X	Y	Z

UNDERSTANDING SOLUTION

- Do all executions of Gale-Shapley yield same stable matching?
- Man *m* is “valid partner” for woman *w* if some stable matching exists where they are matched
- Man-Optimal Assignment
  - One in which every man receives best valid partner
- Claim 1
  - All executions of GS yield man-optimal assignment
- Claim 2
  - All executions of GS yield woman-pessimal assignment (each woman receives worst possible valid partner)

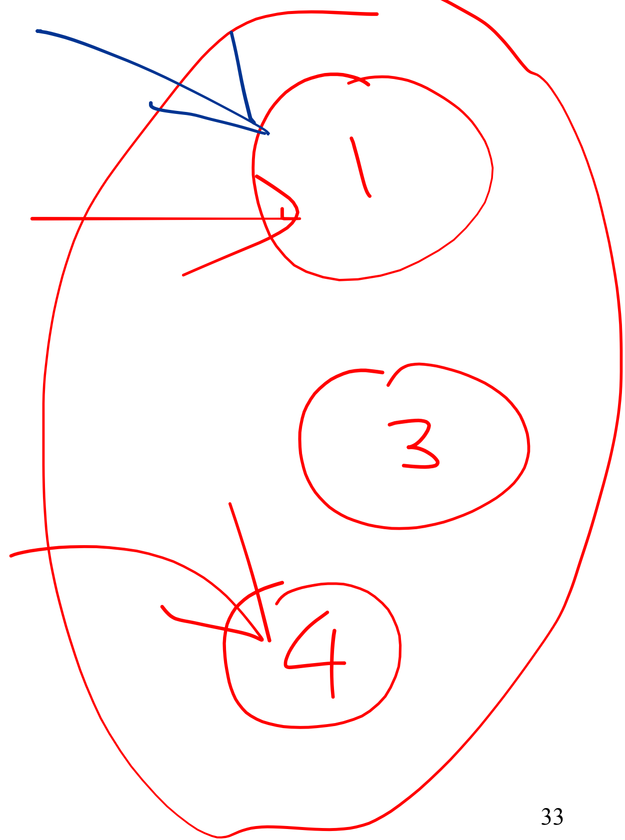
1

2

3

4

5



SIMILAR PROBLEMS

☞ Consider Stable Roommate Problem

- 2*n* people each rank others from 1 to 2*n*-1
- Goal to assign roommate pairs so none unstable

	1st	2nd	3rd
Adam	B	C	D
Bob	C	A	D
Chris	A	B	D
Doofus	A	B	C

A-B, C-D $\Rightarrow$ B-C unstable
A-C, B-D $\Rightarrow$ A-B unstable
A-D, B-C $\Rightarrow$ A-C unstable

☞ Observation

- Stable matching for stable roommate problem doesn't always exist

FINAL THOUGHTS

- ☞ Steps in Algorithm Design
  - Formulate problem precisely
  - Design algorithm for problem
  - Prove algorithm correct
  - Give bound on algorithm’s running time
- ☞ Design Techniques
  - This class will explore algorithm design by enumerating set of design techniques
    - Will learn to recognize problems likely to belong to one class or another

EXAMPLE

☞ If all men have same list of preferences and all women have same list of preferences

- Prove only one stable matching exists

Summary

☞ **Stable Marriage problem an archetype of many matching problems**

☞ **Illustrates**

- **Development of algorithm through intuition**
- **Proof of algorithm**
- **Analysis of time complexity**
- **Application of algorithm to real-life situations**
- **Adaptation of algorithm to solving related problems**