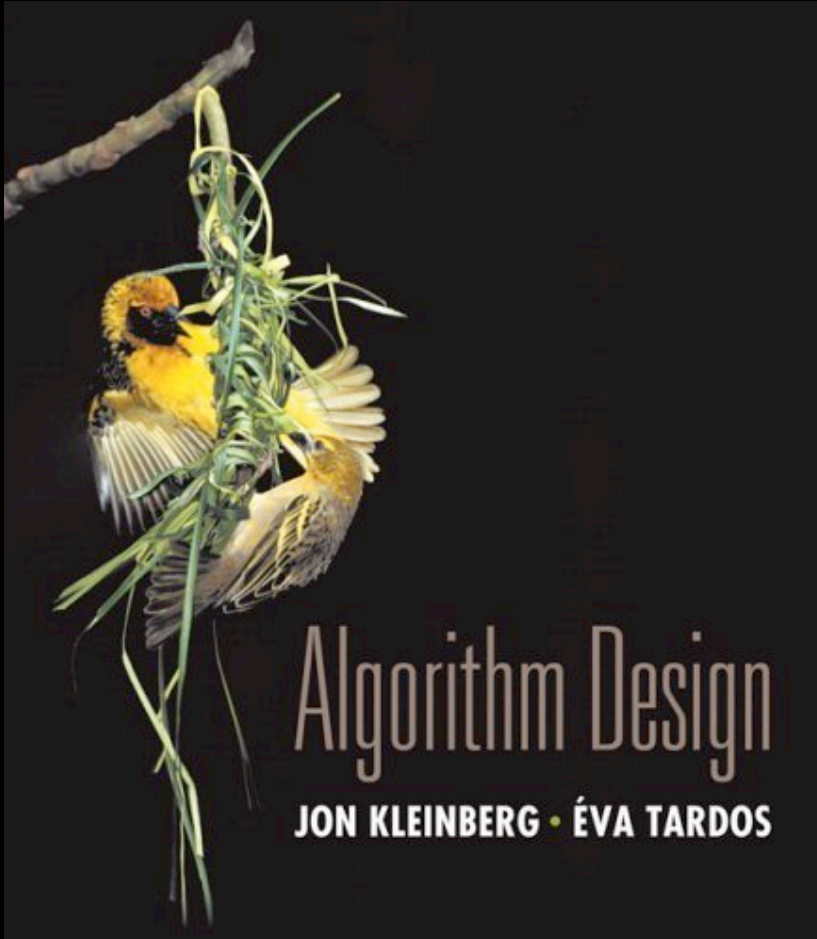


Chapter 1

Introduction: Some Representative Problems



Slides by Kevin Wayne.

Copyright © 2005 Pearson-Addison Wesley.

All rights reserved.

1.1 A First Problem: Stable Matching

Matching Residents to Hospitals

Goal. Given a set of preferences among hospitals and medical school students, design a **self-reinforcing** admissions process.

Unstable pair: applicant x and hospital y are **unstable** if:

- x prefers y to its assigned hospital.
- y prefers x to one of its admitted students.

Stable assignment. Assignment with no unstable pairs.

- Natural and desirable condition.
- Individual self-interest will prevent any applicant/hospital deal from being made.

Stable Matching Problem

Goal. Given n men and n women, find a "suitable" matching.

- Participants rate members of opposite sex.
- Each man lists women in order of preference from best to worst.
- Each woman lists men in order of preference from best to worst.

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

Men's Preference Profile

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

Women's Preference Profile

Stable Matching Problem

Perfect matching: everyone is matched monogamously.

- Each man gets exactly one woman.
- Each woman gets exactly one man.

Stability: no incentive for some pair of participants to undermine assignment by joint action.

- In matching M , an unmatched pair m - w is **unstable** if man m and woman w prefer each other to current partners.
- Unstable pair m - w could each improve by eloping.

Stable matching: perfect matching with no unstable pairs.

Stable matching problem. Given the preference lists of n men and n women, find a stable matching if one exists.

Stable Matching Problem

Q. Is assignment X-C, Y-B, Z-A stable?

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

Men's Preference Profile

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

Women's Preference Profile

Stable Matching Problem

Q. Is assignment X-C, Y-B, Z-A stable?

A. No. Bertha and Xavier will hook up.

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

Men's Preference Profile

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

Women's Preference Profile

Stable Matching Problem

Q. Is assignment X-A, Y-B, Z-C stable?

A. Yes.

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

Men's Preference Profile

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

Women's Preference Profile

Stable Roommate Problem

Q. Do stable matchings always exist?

A. Not obvious a priori.

↖ is core of market nonempty?

Stable roommate problem.

- $2n$ people; each person ranks others from 1 to $2n-1$.
- Assign roommate pairs so that no unstable pairs.

	1 st	2 nd	3 rd
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 matchings do not always exist for stable roommate problem.

Propose-And-Reject Algorithm

Propose-and-reject algorithm. [Gale-Shapley 1962] Intuitive method that guarantees to find a stable matching.

```
Initialize each person to be free.  
while (some man is free and hasn't proposed to every woman) {  
    Choose such a man m  
    w = 1st woman on m's list to whom m has not yet proposed  
    if (w is free)  
        assign m and w to be engaged  
    else if (w prefers m to her fiancé m')  
        assign m and w to be engaged, and m' to be free  
    else  
        w rejects m  
}
```

Proof of Correctness: Termination

Observation 1. Men propose to women in decreasing order of preference.

Observation 2. Once a woman is matched, she never becomes unmatched; she only "trades up."

Claim. Algorithm terminates after at most n^2 iterations of while loop.

Pf. Each time through the while loop a man proposes to a new woman. There are only n^2 possible proposals. ■

	1 st	2 nd	3 rd	4 th	5 th
Victor	A	B	C	D	E
Wyatt	B	C	D	A	E
Xavier	C	D	A	B	E
Yancey	D	A	B	C	E
Zeus	A	B	C	D	E

	1 st	2 nd	3 rd	4 th	5 th
Amy	W	X	Y	Z	V
Bertha	X	Y	Z	V	W
Clare	Y	Z	V	W	X
Diane	Z	V	W	X	Y
Erika	V	W	X	Y	Z

$n(n-1) + 1$ proposals required

Proof of Correctness: Perfection

Claim. All men and women get matched.

Pf. (by contradiction)

- Suppose, for sake of contradiction, that Zeus is not matched upon termination of algorithm.
- Then some woman, say Amy, is not matched upon termination.
- By Observation 2, Amy was never proposed to.
- But, Zeus proposes to everyone, since he ends up unmatched. ■

Proof of Correctness: Stability

Claim. No unstable pairs.

Pf. (by contradiction)

- Suppose A - Z is an unstable pair: each prefers each other to partner in Gale-Shapley matching S^* .

- Case 1: Z never proposed to A .
 $\Rightarrow Z$ prefers his GS partner to A .
 $\Rightarrow A$ - Z is stable.
 men propose in decreasing order of preference
- Case 2: Z proposed to A .
 $\Rightarrow A$ rejected Z (right away or later)
 $\Rightarrow A$ prefers her GS partner to Z .
 $\Rightarrow A$ - Z is stable. \leftarrow women only trade up

- In either case A - Z is stable, a contradiction. ■

S^*

Amy-Yancey
Bertha-Zeus
...

Summary

Stable matching problem. Given n men and n women, and their preferences, find a stable matching if one exists.

Gale-Shapley algorithm. Guarantees to find a stable matching for **any** problem instance.

Q. How to implement GS algorithm efficiently?

Q. If there are multiple stable matchings, which one does GS find?

Efficient Implementation

Efficient implementation. We describe $O(n^2)$ time implementation.

Representing men and women.

- Assume men are named $1, \dots, n$.
- Assume women are named $1', \dots, n'$.

Engagements.

- Maintain a list of free men, e.g., in a queue.
- Maintain two arrays `wife[m]`, and `husband[w]`.
 - set entry to 0 if unmatched
 - if m matched to w then `wife[m]=w` and `husband[w]=m`

Men proposing.

- For each man, maintain a list of women, ordered by preference.
- Maintain an array `count[m]` that counts the number of proposals made by man m .

Efficient Implementation

Women rejecting/accepting.

- Does woman w prefer man m to man m' ?
- For each woman, create an array with the **rank** of each man.
- Constant time access for each query after $O(n)$ preprocessing.

Amy	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th
Pref	8	3	7	1	4	5	6	2

Amy	1	2	3	4	5	6	7	8
Rank	4 th	8 th	2 nd	5 th	6 th	7 th	3 rd	1 st

```
for i = 1 to n
    rank[pref[i]] = i
```

Amy prefers man 3 to 6

since $\text{rank}[3] < \text{rank}[6]$

2

7

Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of *Gale-Shapley* yield the same stable matching? If so, which one?

An instance with two stable matchings.

- A-X, B-Y, C-Z.
- A-Y, B-X, C-Z.

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

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

Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of *Gale-Shapley* yield the same stable matching? If so, which one?

Def. Man m is a **valid partner** of woman w if there exists some stable matching in which they are matched.

Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of *Gale-Shapley* yield the same stable matching? If so, which one?

Def. Man m is a **valid partner** of woman w if there exists some stable matching in which they are matched.

Man-optimal assignment. Each man receives best valid partner.

Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?

Def. Man m is a **valid partner** of woman w if there exists some stable matching in which they are matched.

Man-optimal assignment. Each man receives best valid partner.

Claim. All executions of GS yield **man-optimal** assignment, which is a stable matching!

- No reason a priori to believe that man-optimal assignment is perfect, let alone stable.
- Simultaneously best for each and every man.

Man Optimality

Claim. GS matching S^* is man-optimal.

Pf. (by contradiction)

- Suppose some man is paired with someone other than best partner. Men propose in decreasing order of preference \Rightarrow some man is rejected by valid partner.
- Let Y be **first** such man, and let A be **first** valid woman that rejects him.
- Let S be a stable matching where A and Y are matched.
- When Y is rejected, A forms (or reaffirms) engagement with a man, say Z , whom she prefers to Y .
- Let B be Z 's partner in S .
- Z not rejected by any valid partner at the point when Y is rejected by A . Thus, Z prefers A to B .
- But A prefers Z to Y .
- Thus A - Z is unstable in S . ■

S
Yancey-Amy
Zeus-Bertha
...

↑
since this is first rejection
by a valid partner

Stable Matching Summary

Stable matching problem. Given preference profiles of n men and n women, find a **stable** matching.

↙
no man and woman prefer to be with each other than assigned partner

Gale-Shapley algorithm. Finds a stable matching in $O(n^2)$ time.

Man-optimality. In version of *GS* where men propose, each man receives best valid partner.

↙
 w is a valid partner of m if there exist some stable matching where m and w are paired

Q. Does man-optimality come at the expense of the women?

Woman Pessimality

Woman-pessimal assignment. Each woman receives worst valid partner.

Claim. GS finds **woman-pessimal** stable matching S^* .

Pf.

- Suppose A - Z matched in S^* , but Z is not worst valid partner for A .
- There exists stable matching S in which A is paired with a man, say Y , whom she likes less than Z .
- Let B be Z 's partner in S .
- Z prefers A to B . — man-optimality
- Thus, A - Z is an unstable in S . ■

S	
	Amy-Yancey
	Bertha-Zeus
	...

Extensions: Matching Residents to Hospitals

Ex: Men \approx hospitals, Women \approx med school residents.

Variant 1. Some participants declare others as unacceptable.

Variant 2. Unequal number of men and women.

↖
resident A unwilling to
work in Cleveland

Variant 3. Limited polygamy.

↖
hospital X wants to hire 3 residents

Def. Matching S **unstable** if there is a hospital h and resident r such that:

- h and r are acceptable to each other; and
- either r is unmatched, or r prefers h to her assigned hospital; and
- either h does not have all its places filled, or h prefers r to at least one of its assigned residents.

Application: Matching Residents to Hospitals

NRMP. (National Resident Matching Program)

- Original use just after WWII. — predates computer usage
- Ends of March, 23,000+ residents.

Rural hospital dilemma.

- Certain hospitals (mainly in rural areas) were unpopular and declared unacceptable by many residents.
- Rural hospitals were under-subscribed in NRMP matching.
- How can we find stable matching that benefits "rural hospitals"?

Rural Hospital Theorem (1986). Any hospital that has some empty positions at some stable matching is assigned precisely the same set of residents at every stable matching.

Deceit: Machiavelli Meets Gale-Shapley

Q. Can there be an incentive to misrepresent your preference profile?

- Assume you know men's propose-and-reject algorithm will be run.
- Assume that you know the preference profiles of all other participants.

Fact. No, for any man yes, for some women. No mechanism can guarantee a stable matching and be cheatproof.

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

Men's Preference List

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

Women's True Preference Profile

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

Amy Lies

Lessons Learned

Lessons Learned

Powerful ideas learned in course.

Lessons Learned

Powerful ideas learned in course.

- Isolate underlying structure of problem.

Lessons Learned

Powerful ideas learned in course.

- Isolate underlying structure of problem.
- Create useful and efficient algorithms.

Lessons Learned

Powerful ideas learned in course.

- Isolate underlying structure of problem.
- Create useful and efficient algorithms.

Lessons Learned

Powerful ideas learned in course.

- Isolate underlying structure of problem.
- Create useful and efficient algorithms.

Potentially deep social ramifications. [legal disclaimer]