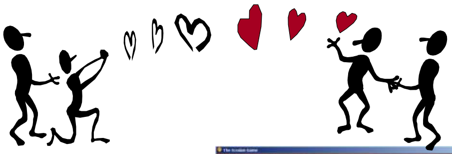
Great Theoretical Ideas In Computer Science

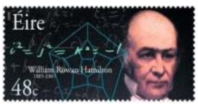
Victor Adamchik

CS 15-251

Carnegie Mellon University

Graphs - III





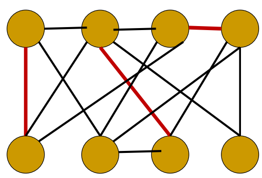
Outline

- Vertex Cover
- Stable Matching
- Gale-Shapely theorem
- Euler Cycle
- Hamiltonian Cycle

Matching

Definition. A subset of edges is a matching if no two edges have a common vertex (mutually disjoint).

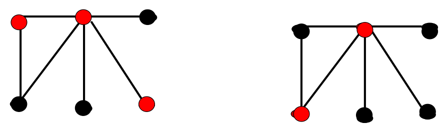
Definition. A maximum matching is a matching with the largest possible number of edges



The Hungarian algorithm for bipartite graphs, and the Blossom algorithm for general graphs.

Vertex Cover

This is a set of vertices such that each edge of the graph is incident to at least one vertex of the set.



A minimum vertex cover is a vertex cover of smallest possible size. A NP-hard problem.

Application: monitoring a worm propagation.

Max-matching vs. Min-cover

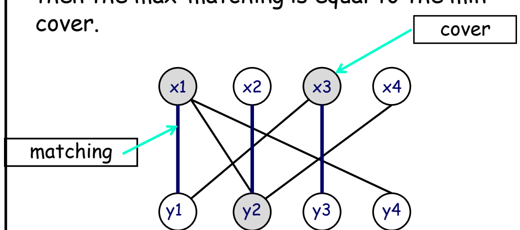
Theorem. The largest number of edges in a matching does not exceed the smallest number of vertices in a cover

Proof.
Since each vertex in a cover C incident with at most one edge in matching M, then $C \geq M$.

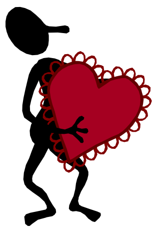
max-matching is in P
min-cover is in NP

Max-matching vs. Min-cover

Theorem (König, 1931). If G is a bipartite graph, then the max-matching is equal to the min-cover.



~~Dating for Computer Scientists~~



The Stable Marriage Problem

There are n men and n women.

Each one has a complete ordered preference list for those of the other sex.

Men's preferences	Women's preferences
1 - 2 4 1 3	1 - 2 1 4 3
2 - 3 1 4 2	2 - 4 3 1 2
3 - 2 3 1 4	3 - 1 4 3 2
4 - 4 1 3 2	4 - 2 1 4 3

The goal is to pair the men with the women.

Does a matching exist?

What criteria to use?

Does a matching exist?

It's a complete bipartite graph

every man-woman edge is present

so clearly there's a perfect matching

by Hall's Theorem

What criteria to use?

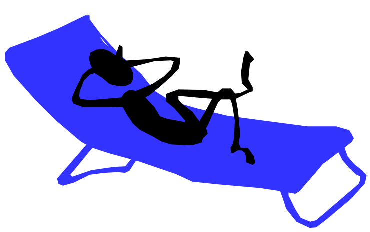
More Than One Notion of What Constitutes A "Good" Pairing

Maximizing total satisfaction

Maximizing the minimum satisfaction

Minimizing maximum difference in mate ranks

Maximizing people who get their first choice



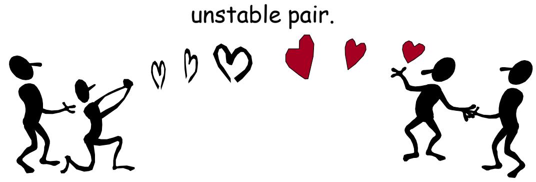
We will ignore the issue of what is "best"!

Unstable Pair

Suppose we pair off all the men and women.

A pair (M,W) is unstable if M and W like each other more than their assigned partners.

A matching is called unstable if it has a unstable pair.



Stable Matchings

A matching of men and women is called stable if it contains no unstable pairs.

Men's preferences	Women's preferences
1 – 3 1 2	1 – 3 2 1
2 – 3 2 1	2 – 2 1 3
3 – 1 2 3	3 – 2 3 1

Stable matching: (1,2) (2,3) (3,1)

Unstable matching: (1,3) (2,2) (3,1)

Find the unstable pair (2,3)

National Residency Match

- About 4000 hospitals try to fill 20,000+ positions.
- Students apply and interview at hospitals in the fall.
- In February, students and hospitals state their preferences
  - Each student submits rank-order list of hospitals
  - Each hospital submits rank-order list of students
- Computer algorithm generates an assignment.

Matching applications galore!

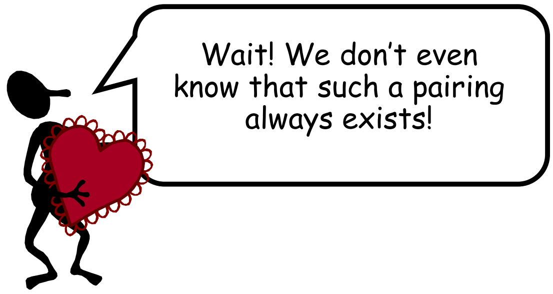
What are the common features of these problems?

- Two sides of the market to be matched.
- Participants on one or both sides care about to whom they are matched.
- For whatever reason, money cannot be used to determine the assignment.

Examples

- Housing assignment
- Fraternity/sorority rush
- MBA course allocation
- Dating websites
- College admissions
- Judicial clerkships
- Military postings
- NCAA football bowls

Given a set of preference lists, how do we find a stable matching?



Gale-Shapely Theorem (1965)

Theorem.
Stable matching is always possible.

We will prove this theorem by presenting an algorithm that always returns a stable matching.

Basic principle: Man proposes, woman disposes

Algorithm: a general idea

Each unattached man proposes to the highest-ranked woman in his list, who has not already rejected him.

If the man proposing to her is better than her current mate, the woman dumps her current partner, and becomes engaged to the new proposer.

Continue until all men are attached, and we get a stable matching.

Stable Matching Algorithm

LIST={1,…,n}: list of unattached men.

cur(m): highest ranked woman in m's list, who has not rejected him.

Take a man, say, Bob, from the LIST.

Bob proposes to Alice, where Alice = cur(Bob).

If Alice unattached, Bob and Alice are engaged.

If Alice is engaged to, say, John, but prefers Bob, she dumps John, and Bob and Alice are engaged. Otherwise, she rejects Bob.

The rejected man rejoins LIST, and updates his cur.

Stop when LIST is empty.

Exercise

Given these preferences. Find a stable matching.

Men's preferences

1	2	4	1	3
2	3	1	4	2
3	2	3	1	4
4	4	1	3	2

Women's preferences

1	2	1	4	3
2	4	3	1	2
3	1	4	3	2
4	2	1	4	3

Man proposes, woman disposes

Exercise

Given these preferences. Find a stable matching.

Men's preferences

1	2	4	1	3
2	3	1	4	2
3	2	3	1	4
4	4	1	3	2

Women's preferences

1	2	1	4	3
2	4	3	1	2
3	1	4	3	2
4	2	1	4	3

Algorithm

1	~~2~~	4	
2	3		
3	2		
4	~~4~~		1

For any given instance of the stable marriage problem, the Gale-Shapley algorithm terminates, and, on termination, the engaged pairs constitute a stable matching.

First we make the following simple observations:

(1) The engagement always forms a matching.

(2) Once a woman is engaged, she remains engaged.

(3) Each new engagement is a better man for her.

A man cannot be rejected by all women. Because if he is, then all women must be engaged, which is impossible.

The algorithm terminates in at most n^2 iterations.

Remains to prove the matching is always stable.

Why is it stable?

Suppose the resulting matching has a unstable pair (Mark, Laura).

Mark must have proposed to Laura at some point.

During the algorithm, Laura also rejected Mark in favor of some she prefers more.

Laura's current partner must be more desirable than Mark.

Thus, the pair (Mark, Laura) is not unstable.

Question about this algorithm

Notice that our algorithm is asymmetrical. It does not treat men and women the same way.

Do you think this algorithm is better for the men or for the women?

If you reverse the roles, would it lead to better results for the women, or the men?

Answer: The algorithm is better for the men. In fact it's the best possible.

This algorithm matches every man with his best woman!

Thus, this is the man-optimal algorithm.

Theorem: Gale-Shapley algorithm finds MAN-OPTIMAL stable matching!

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

Man-optimal assignment: every man receives best valid partner.

Theorem: Gale-Shapley algorithm finds MAN-OPTIMAL stable matching!

M
Tom-Amy
Steve-Lucy

Proof (by contradiction): Let Tom be paired with someone other than his best partner.

Thus he is the first man rejected by a valid partner, say Amy.

When Tom is rejected, Amy gets engaged with Steve. Amy prefers Steve to Tom.

Steve not rejected by any valid partner, since Tom is the first to be rejected by valid partner. Thus, Steve prefers Amy to Lucy. But Amy prefers Steve to Tom.

Thus (Amy, Steve) is unstable pair in M.

Theorem:

Gale-Shapley algorithm finds WOMAN-PESSIMAL matching.

Each woman married to the worst valid partner.

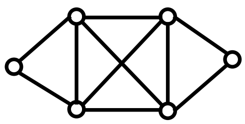
How would you fix this?

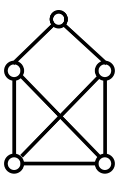
Woman proposes, man disposes

Euler and Hamiltonian cycles

The Euler cycle

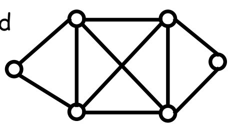
Is it possible to traverse each of the edges of a graph exactly once, starting and ending at the same vertex?



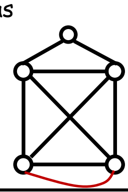


The Euler cycle/path

Theorem. A connected undirected graph has an Euler cycle iff each vertex is of even degree.



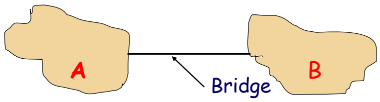
Theorem. A connected undirected graph has an Euler path (not a cycle) iff it has exactly two vertices of odd degree.



The Euler cycle: Fleury's algorithm

The Euler theorem does not tells us how to find that cycle.

The idea behind the algorithm: Don't burn your bridges behind you.

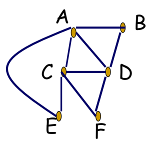


You would only want to cross that bridge if you know that all edges in A have been traveled.

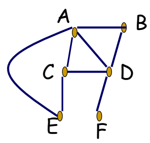
Fleury's Algorithm (1883)

- First make sure the graph is connected and all the vertices have even degree.
- Pick any vertex at random
- When you have a choice, always choose to travel along an edge that is not a bridge of the yet-to-be-traveled part of the graph.
- Remove that edge.
- When you cannot travel any more, stop. You are done.

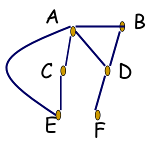
Fleury's Algorithm



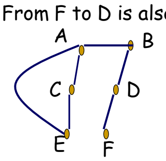
Start at F (arbitrarily)



Travel from F to C
From F to D is also possible

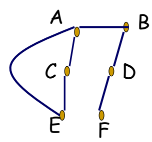


Travel from C to D

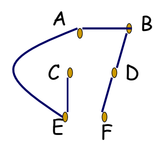


Travel from D to A, but not F

Fleury's Algorithm



Travel from A to C, but not B



The rest of moves is forced

The cycle: F->C->D->A->C->E->A->B->D->F

Fleury's Algorithm

The postponing of the use of bridges is really the critical feature of this algorithm, and its purpose is to avoid becoming trapped in some component of *G*.

Checking for bridges is expensive, so the algorithm might run in $O(E^2)$ time.

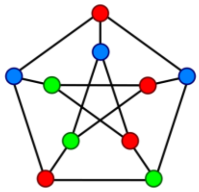
Hierholzer's Algorithm (1883)

- First make sure the graph is connected and all the vertices have even degree.
- Pick any vertex at random
- Find a cycle C_0 using DFS. Remove its edges from the graph
- LOOP: as long as there is an incident edge to some vertex in C_k
    - Find a cycle C_{k+1}
    - Remove edges from the graph
    - Glue two cycles together

This will make a linear time algorithm.

The Hamiltonian cycle

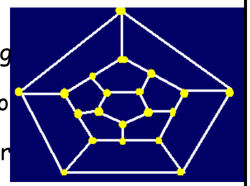
Is it possible to traverse each of the vertices of a graph exactly once, starting and ending at the same vertex?



Does the Petersen graph has a Hamiltonian cycle?

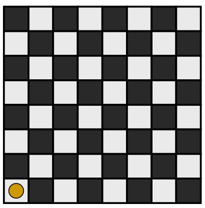
The Hamiltonian cycle

The terminology came from the Icosian puzzle, invented by Hamilton in 1857. There was a wooden dodecahedron with a peg at each vertex labeled with different cities. The goal was to start at a city and travel along edges, visiting each of the other 19 cities exactly ones.



This how sir Hamilton was trying to make some extra money…

The knight's tour problem



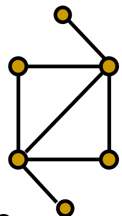
Can a knight visit all squares of a chessboard exactly once, starting at some square, and by making 63 legitimate moves?

The knight's tour problem is a special case of the Hamiltonian circle problem.

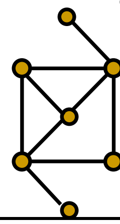
The answer is yes!

Hamilton versus Euler

This figure shows a graph that has a Euler path and has neither Hamiltonian cycle nor Hamiltonian paths.



This one is nether Eulerian not Hamiltonian



Showing that a graph is not Hamiltonian

There are three simple rules that based on observation that any Hamiltonian cycle must contain exactly two edges incident on each vertex.

Rule 1. If a vertex has degree 2, both edges must be on a cycle.

Rule 2. No cycles can be formed until all the vertices have been visited

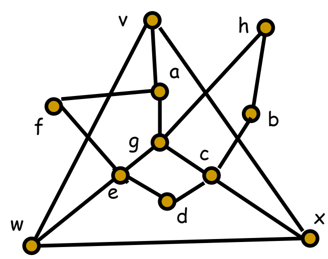
Rule 3. Once we use two edges at a vertex, all other (unused) incident edges must be removed from consideration.

Showing that a Graph is not Hamiltonian

Rule 3 to c:
Delete (c,x), (c,g)

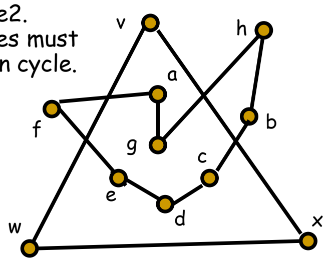
Rule 3 to e:
Delete (e,w), (e,g)

Rule 3 to a:
Delete (a,v)



Showing that a Graph is not Hamiltonian

Each vertex has degree2.
Rule 1 implies that edges must be on every Hamiltonian cycle.



But these form two cycles. Thus, the original graph is not Hamiltonian.

The Hamiltonian cycle

No property is known to efficiently verify existence of a Hamiltonian cycle/path for general graphs.

Here is a sufficient condition:

Theorem: If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex is at least $n/2$, then G has a Hamiltonian cycle.

Theorem: If G is a simple graph with $n \geq 3$ vertices such that the $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices u and v, then G has a Hamiltonian cycle.

Hamiltonian problem is NP

This is a well known NP-complete problem

For general graph, we can not find an exactly linear time complexity algorithm to find a Hamiltonian cycle or path.

However, if such a path exist we can verify it in polynomial time.

Vertex Cover
Stable Matching
Gale-Shapely theorem
Euler Cycle
Hamiltonian Cycle

Here's What You Need to Know…