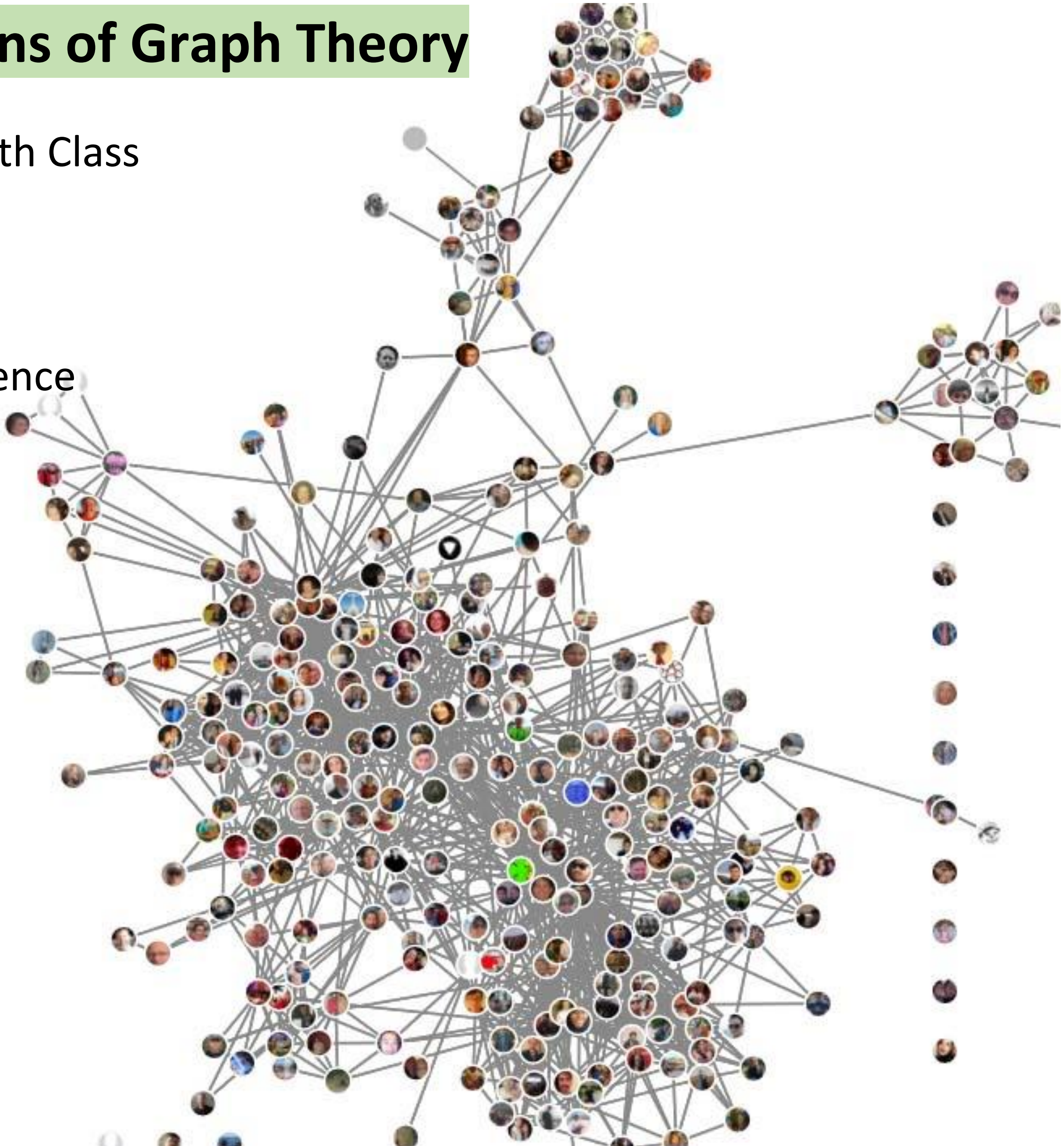
Real-World Applications of Graph Theory

St. John School, 8th Grade Math Class
February 23, 2018

Dr. Dave Gibson, Professor
Department of Computer Science
Valdosta State University



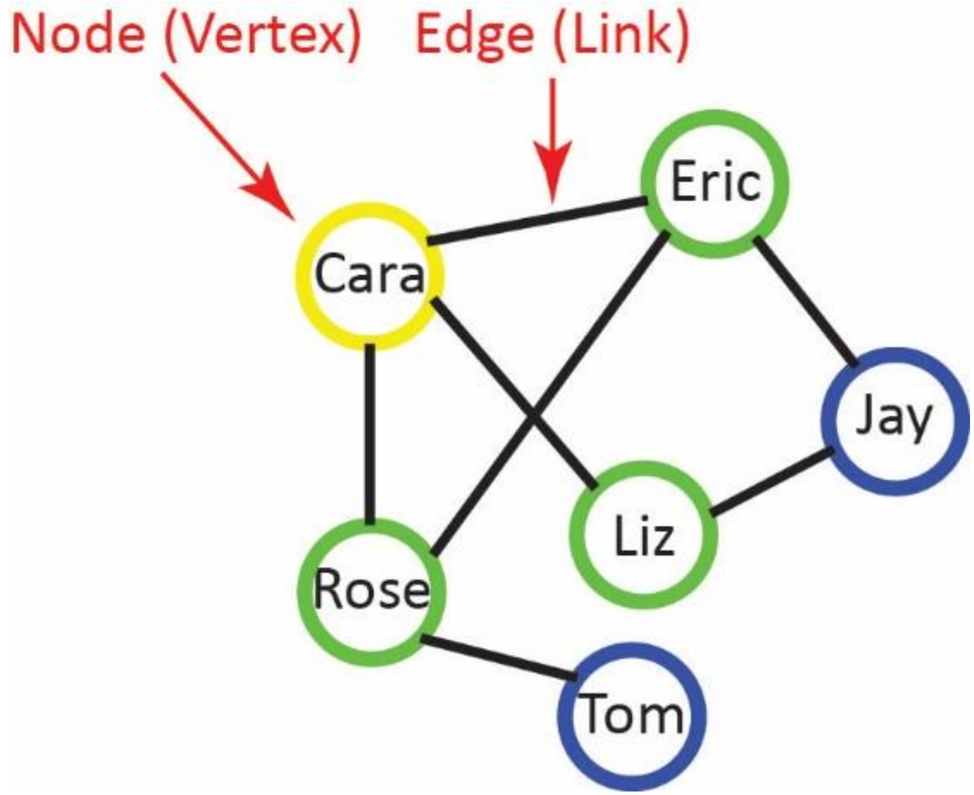
What is a Graph?

- A *graph* is a collection of *nodes* and *edges*. A graph is also called a *network*.

- A *node* is whatever you are interested in: person, city, team, project, computer, *etc*.

- An *edge* represents a relationship between nodes.

- Example: Facebook – the *nodes* are people and the *edges* represent a friend relationship.



- Question – Facebook suggests friends: Who is the first person Facebook should suggest as a friend for Cara? Why?

- *Graph theory* is the study of graphs and is an important branch of computer science and discrete math.

What is an Algorithm?

- All real-world problems are solved with computers.

- Computers can only solve problems if we program it with specific, unambiguous directions.

- An ***algorithm*** is a step-by-step procedure to solve a problem and always give the "best/correct" answer.

- For example, what is an algorithm to solve equations like this for x?

$$4x + 3 = 21 - 2x$$

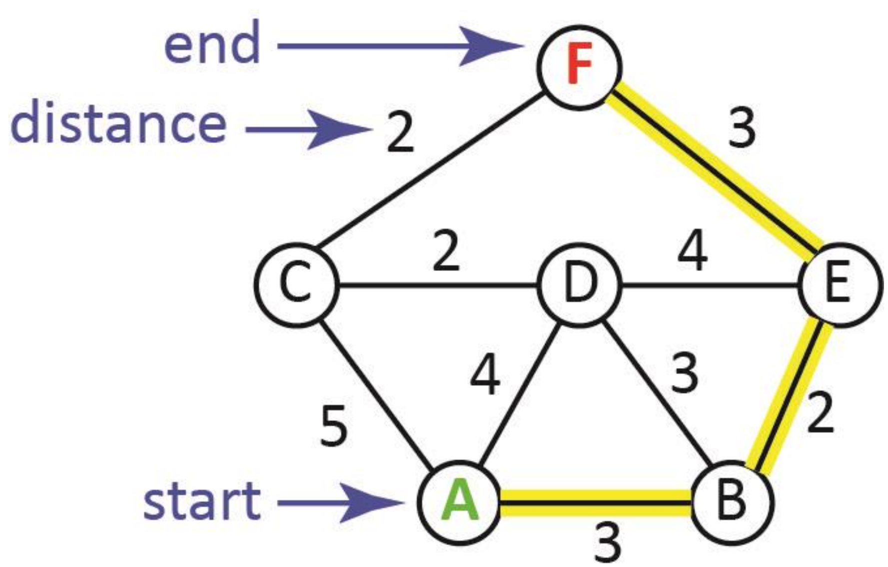
Algorithm

1. Move and all constants to the right side and combine.
2. Move all x's to the left side and combine.
3. Divide the constant on the right by the multiplier of x.

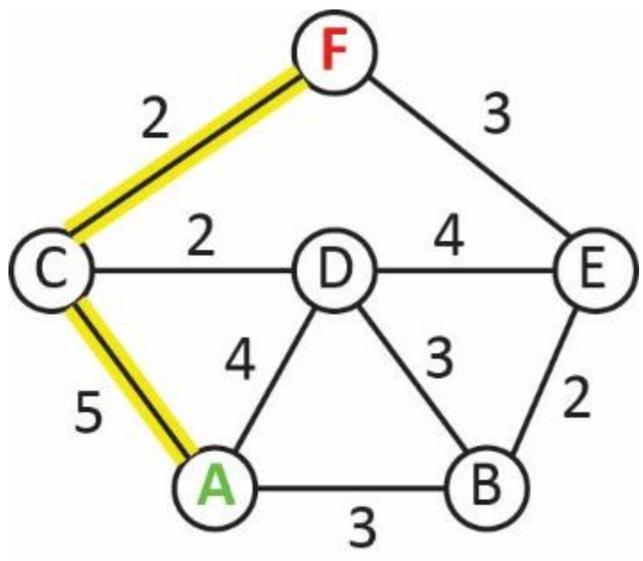
Shortest Path Problem

- What is the shortest path from node A to node F?

A *greedy algorithm* says: "always travel to your nearest neighbor". That doesn't always work.



Because the shortest path is ACF.



Shortest Path Problem

- One solution is *exhaustive search (brute-force)* – which means measuring the total distance of every possible path and then selecting the one with the shortest distance.

- For most real-world problems this is not feasible – there are too many possibilities.

Number of Nodes	Number of Possible Paths	
10	8!=	40,320
20	18!=	6,402,373,705,728,000
30	28!=	304,888,344,611,714,000,000,000,000,000

- There are more than 19,000 cities in the US. How can Google maps calculate the fastest route as quick as you press Enter?

➔ Even with the fastest computers it would take 100's of years to do an exhaustive search

➔ Answer: we have very fast algorithms

Shortest Path Problem

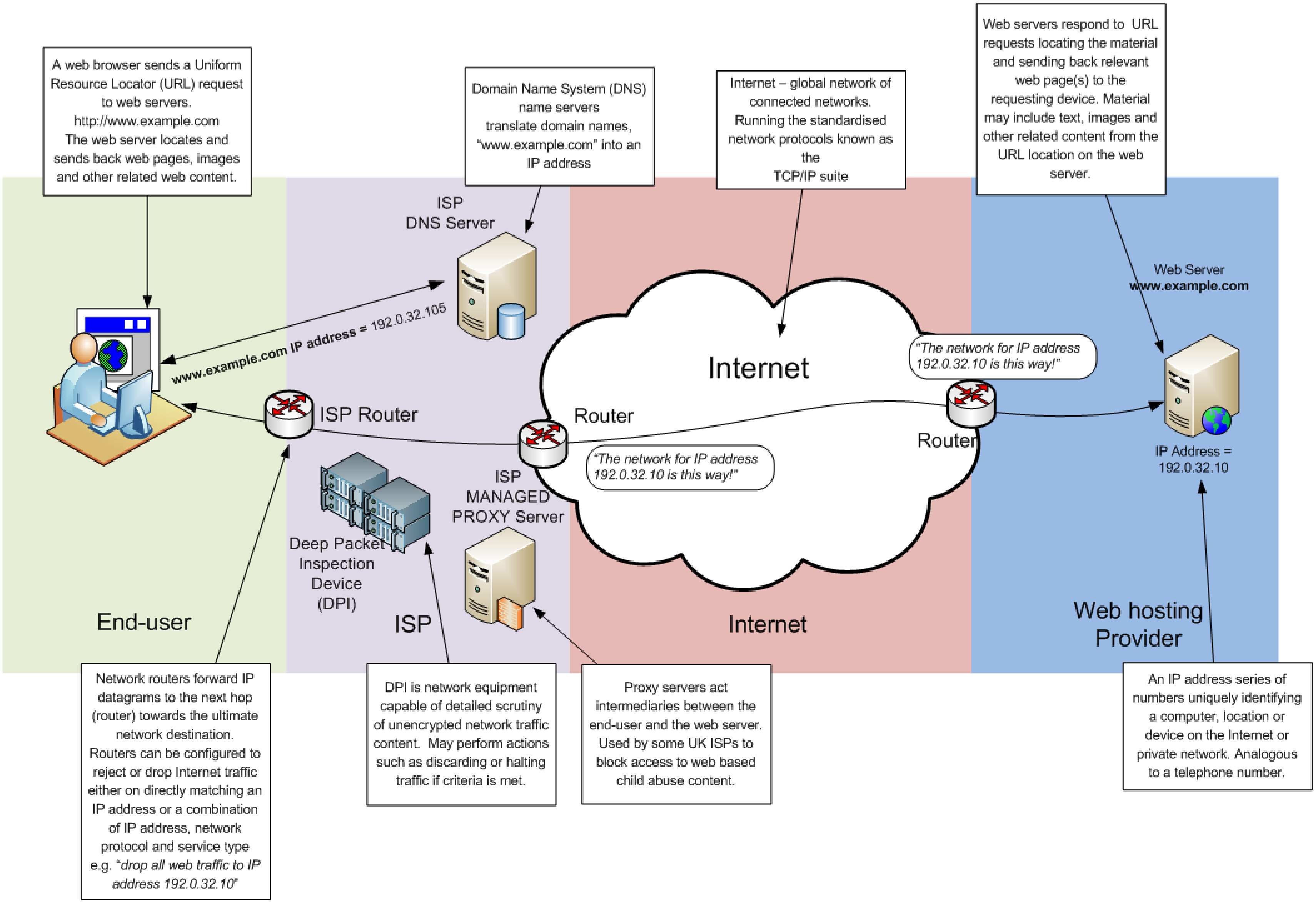
- Dijkstra's algorithm (and others) always finds the best solution extremely quickly. The algorithm is a bit complicated (we won't discuss)

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Mark all nodes unvisited. Set the initial node as current. Create a set of the unvisited nodes called the unvisited set consisting of all the nodes.
3. For the current node, consider all of its unvisited neighbors and calculate their tentative distances. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbor B has length 2, then the distance to B (through A) will be 6 + 2 = 8.
4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.
6. Select the unvisited node that is marked with the smallest tentative distance, and set it as the new "current node" then go back to step 3.

- Countless Applications of the Shortest Path Problem:

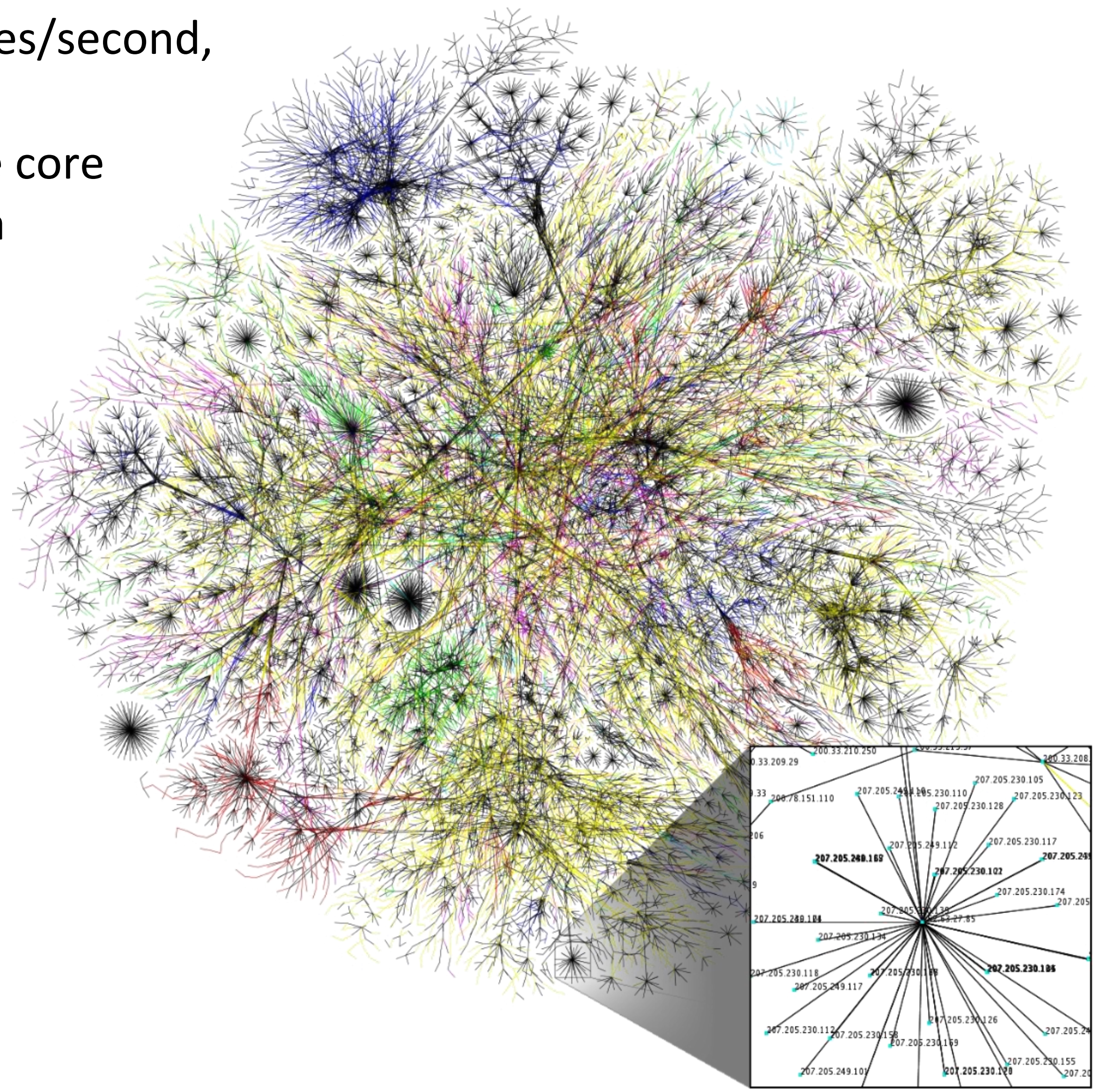
➔ GPS – finding the shortest way to a destination
➔ Routing delivery vehicles, ships, trains, etc
➔ Routing internet traffic
➔ VLSI – design of integrated circuits by combining billions of transistors into a chip
➔ Degree of separation in a social network. The shortest path between two people.

Shortest Path Problem – Routing Internet Traffic



Shortest Path Problem – Routing Internet Traffic

- Google processes 40M searches/second, has 2.5M servers.
- Internet Backbone defines the core routers used to exchange data
- Internet Backbone in Great Britian:



Longest Path – Scheduling and Planning

- Nodes are the tasks that need to be done. Nodes have a “value”, the duration of the task
- Edges represent the dependencies between tasks
- CPM – Critical path method. The longest path is the fastest the project can get done.
- The CP determines the scheduling of tasks and allocation of resources.

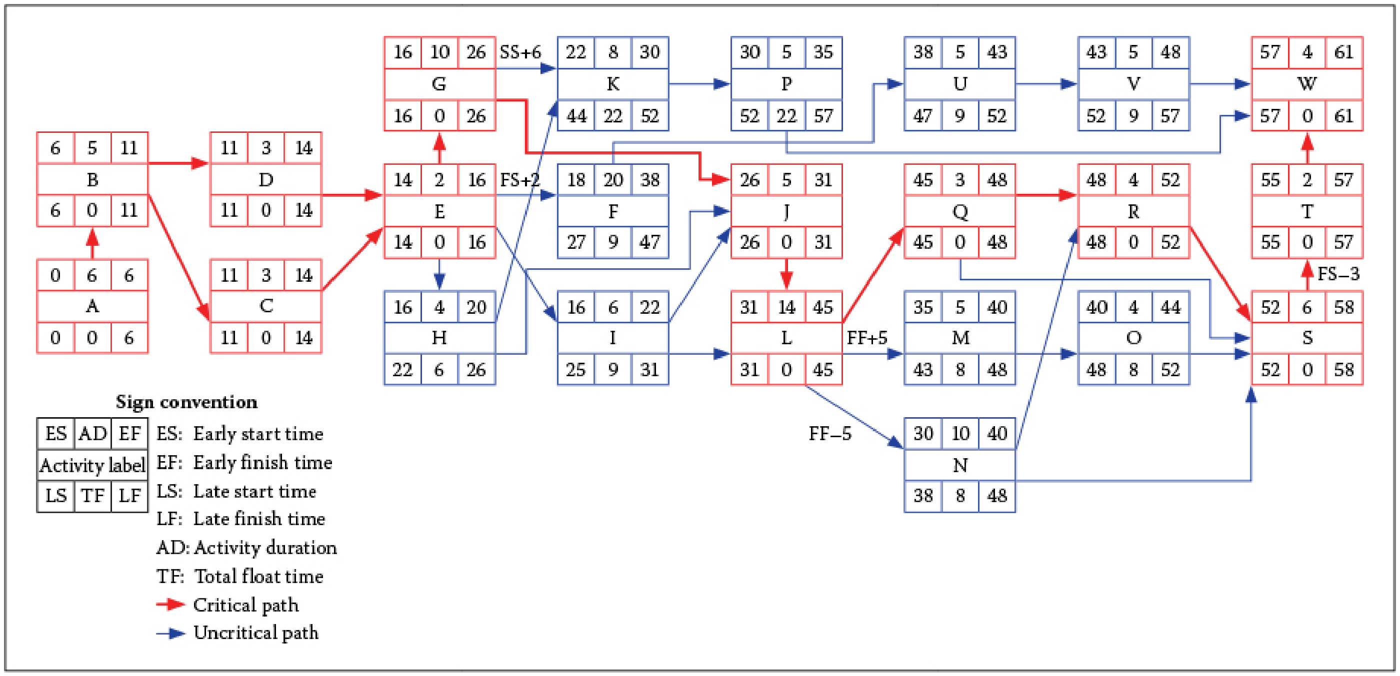
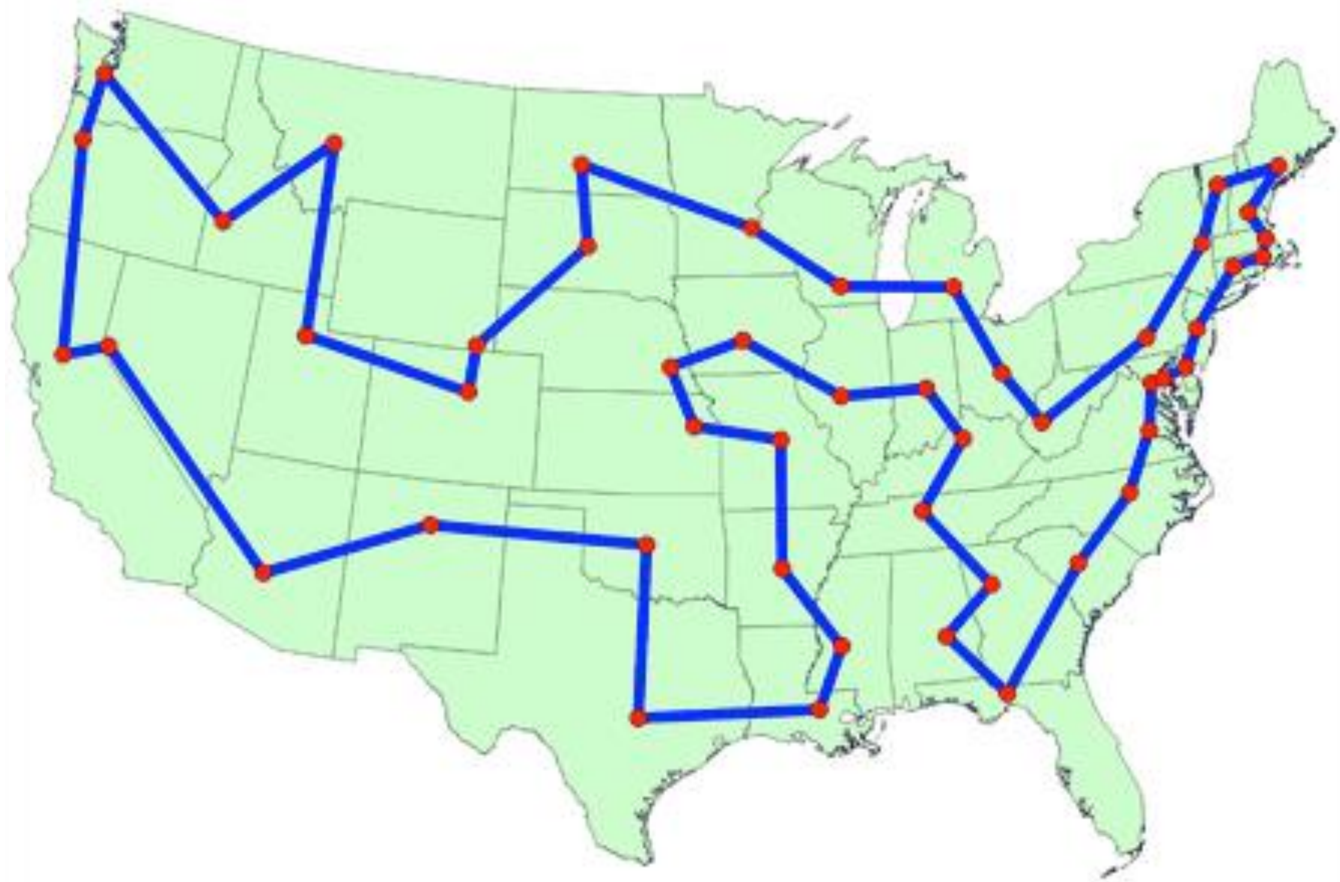


Figure 4 Diagrammatic CPM solution of the single-storey housing project

http://www.scielo.org.za/img/revistas/jsaice/v56n2/02f04.jpg

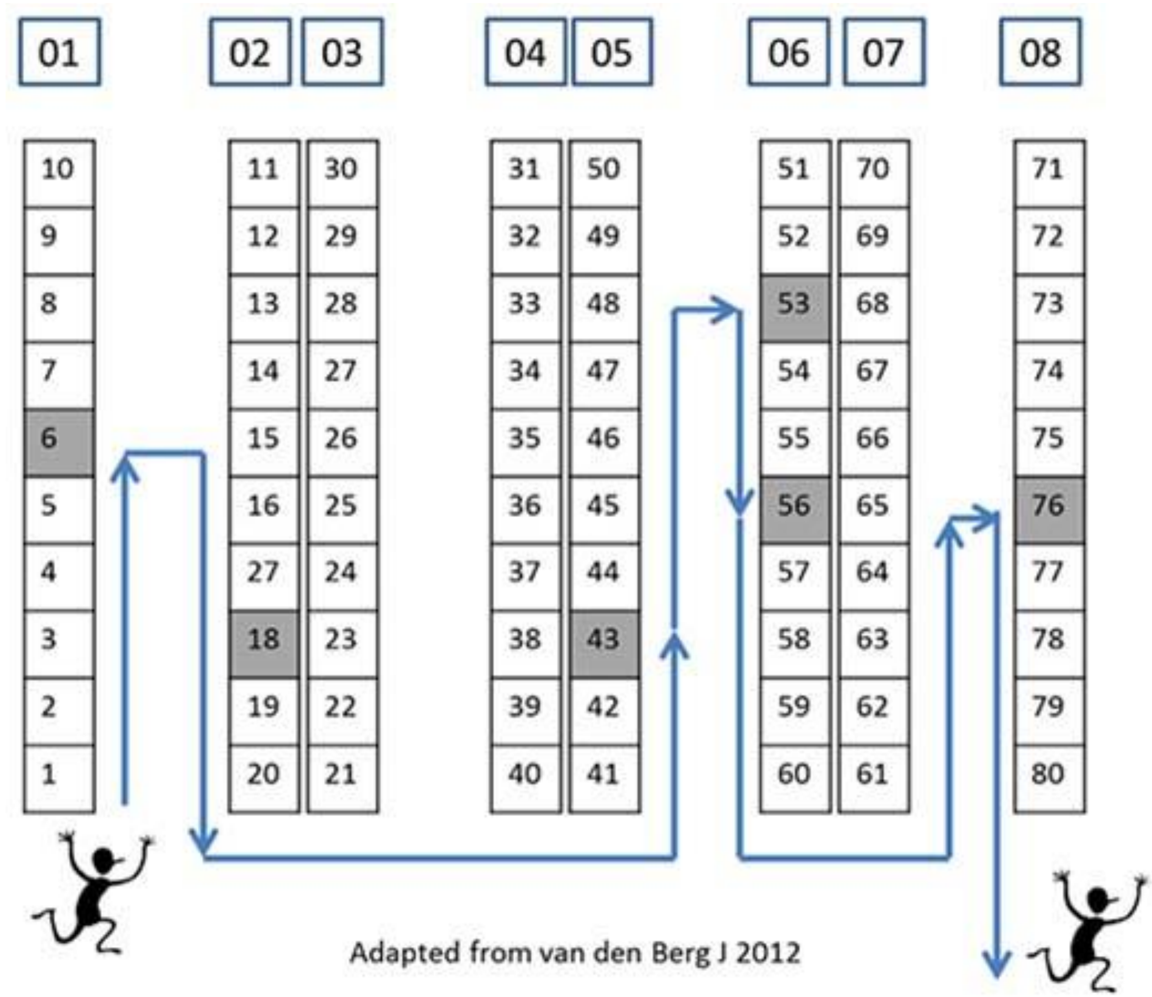
Travelling Salesman Problem (TSP)



- Related to shortest path problem except much more difficult.
- There are no algorithms for TSP!
- Instead, we use <u>*heuristics*</u>.
- A heuristic is the same as an algorithm except the solution is not necessarily the best possible.
- Often with heuristics, we can *bound* the solution we get. For example, a heuristic might guarantee that we are always within 10% of the optimal solution

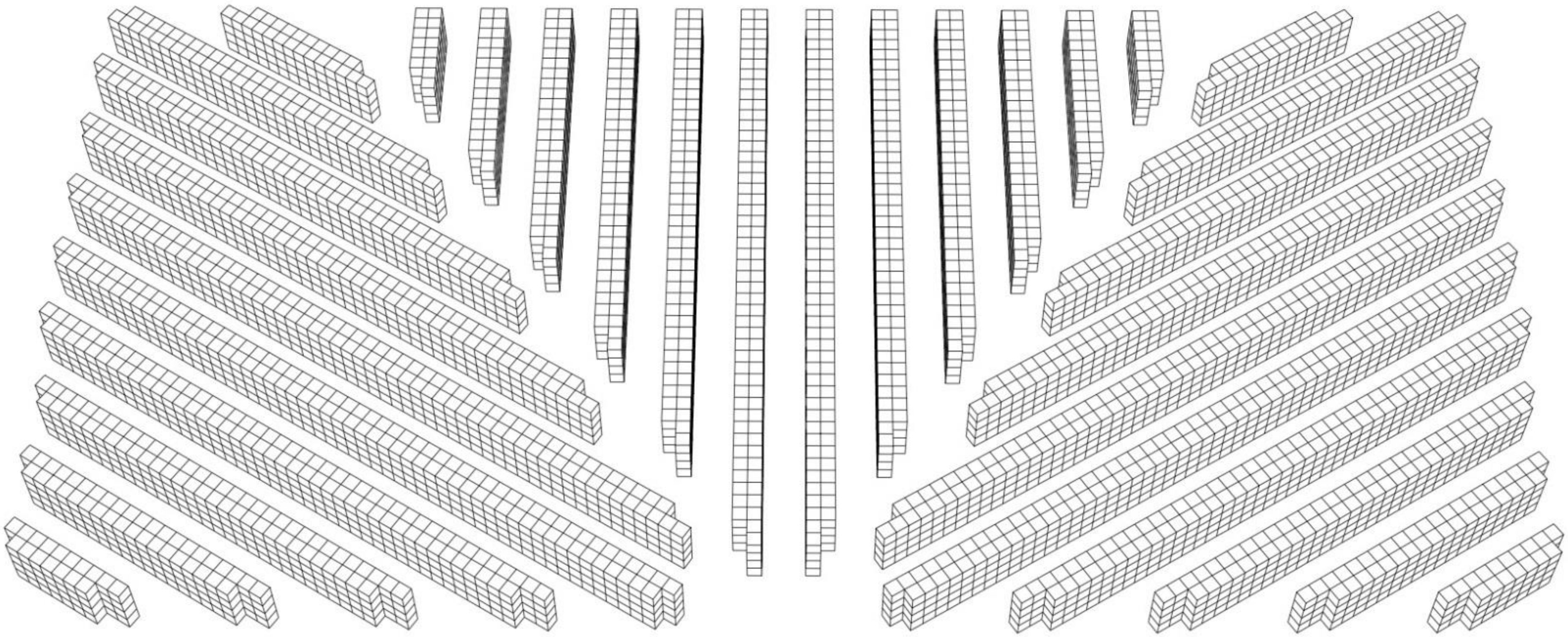
Travelling Salesman Problem (TSP)

- Picking orders in a warehouse



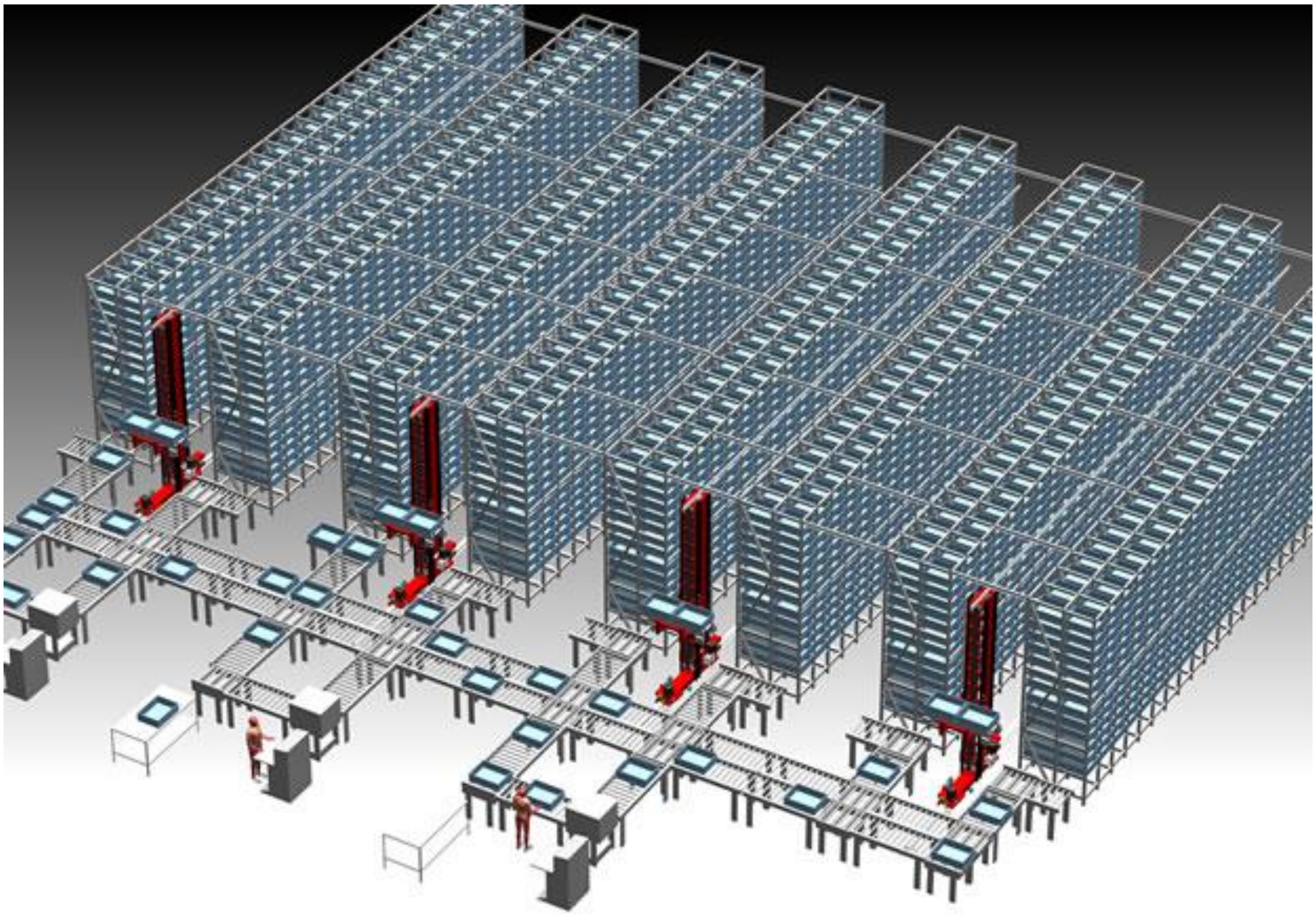
Travelling Salesman Problem (TSP)

- Picking orders in a warehouse – Fishbone Aisle Layout



Travelling Salesman Problem (TSP)

- Picking orders in a warehouse – Automated Storage and Retrieval Systems (ASRS)

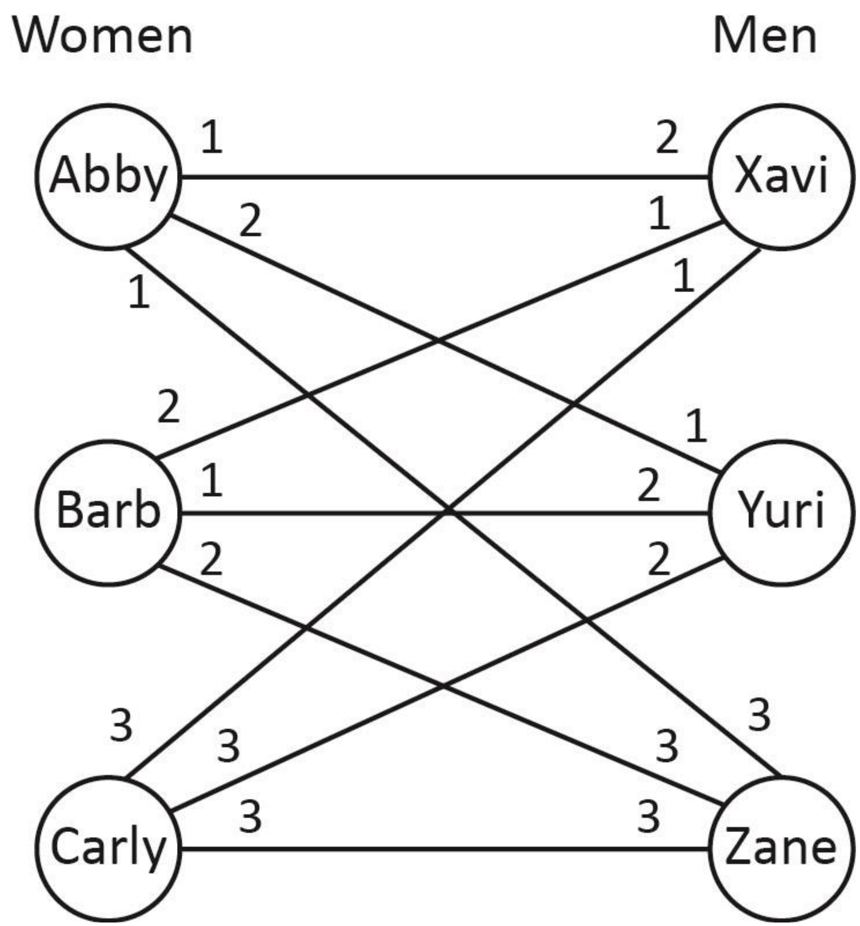


Travelling Salesman Problem (TSP)

- Mass Production – Manufacturing: Drilling holes in sheet metal, rivets, etc

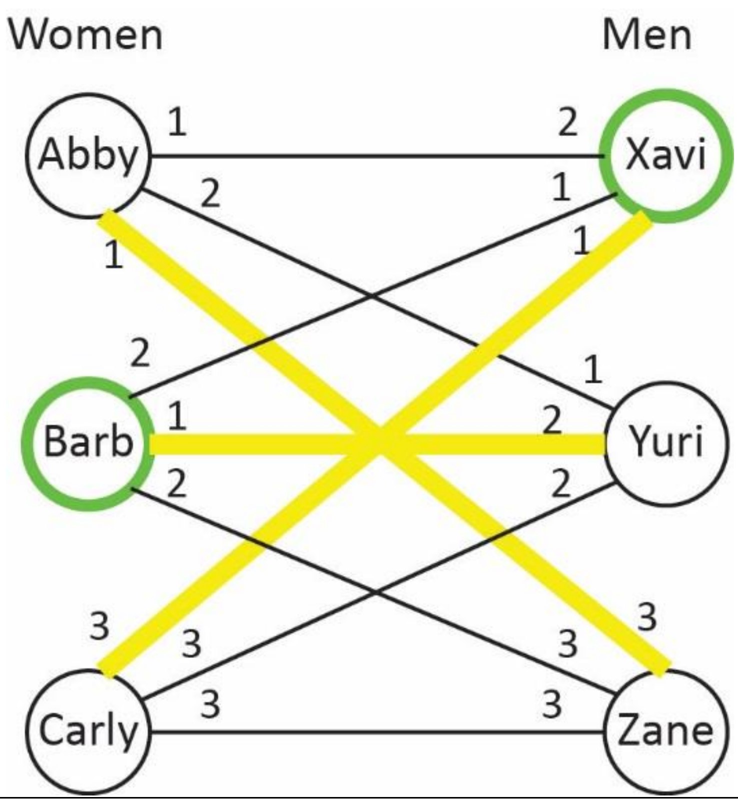
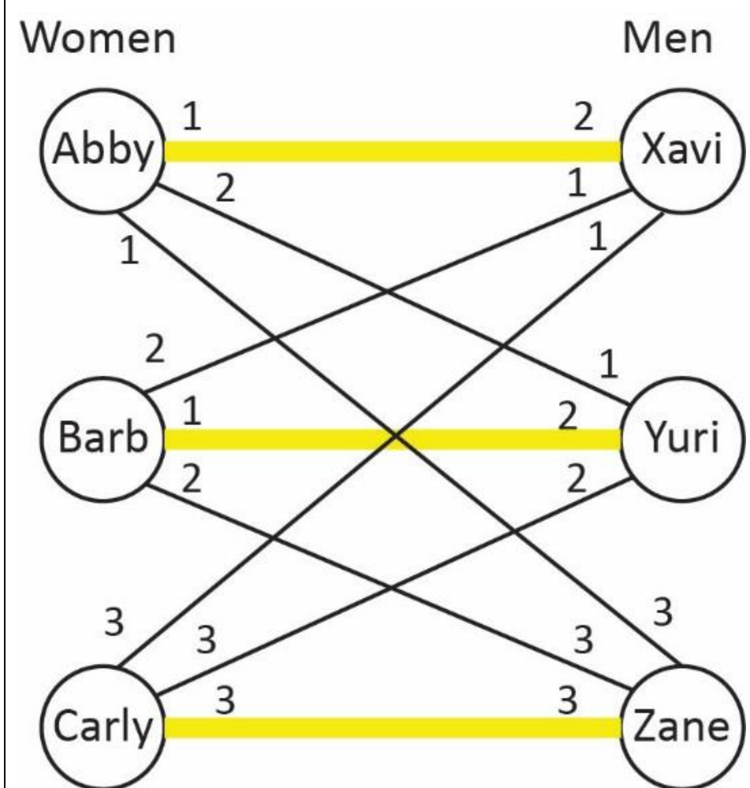
Matching – The Marriage Problem

- Given 3 women and 3 men and their preferences for one another, what is the *best* way to match them?



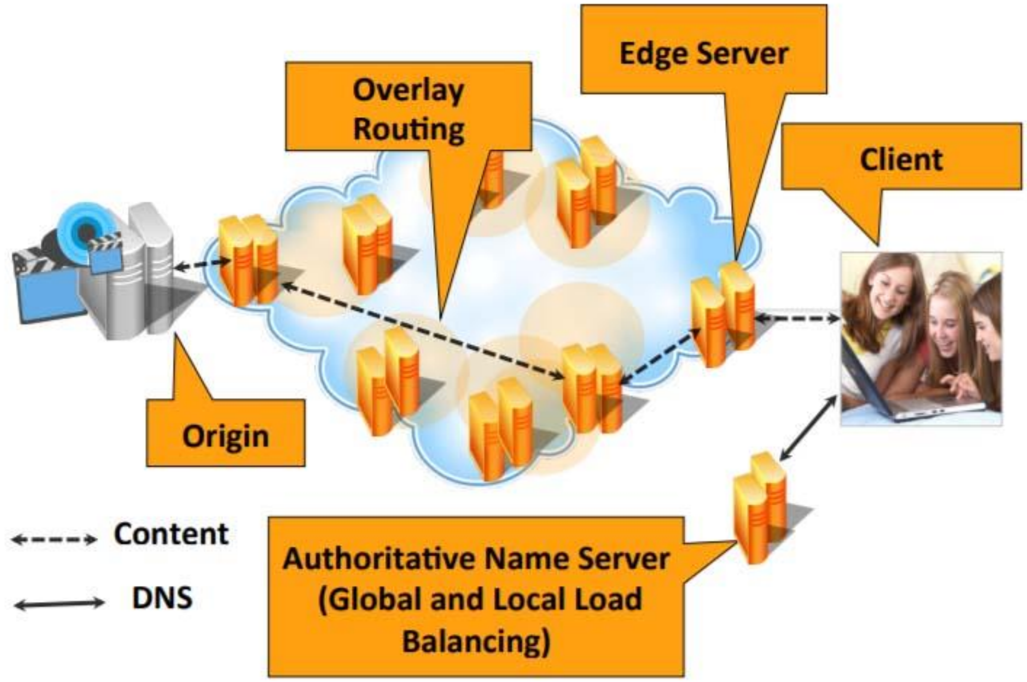
Matching – The Marriage Problem

- One concept in matching is a <u>*stable matching*</u>. A matching is not stable if there exists two people, A and B who are not matched to each other, but both would prefer to be matched to each other.

Unstable Matching		Stable Matching
		

Matching – The Marriage Problem

- Lloyd Shapley and Alvin Roth won the Nobel Prize in Economic Science in 2012 for developing an algorithm to find stable matchings.



- Applications:

➔ Interns (Doctors) to Hospitals. Variation – want to keep couples together.

➔ Organ donation

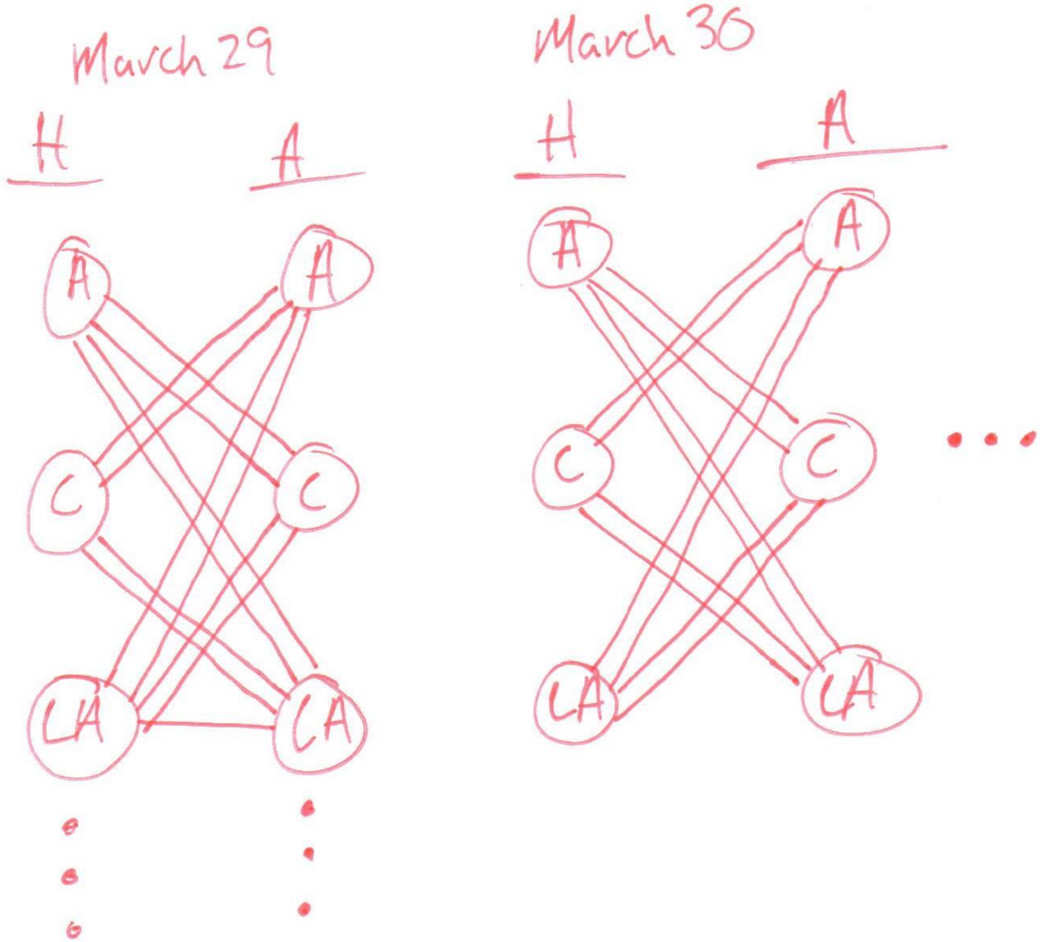
➔ Until 2003 used for applicants to NYC public high schools

➔ Content Delivery Networks – Matching Users to Servers

http://www.sigcomm.org/sites/default/files/ccr/papers/2015/July/0000000-0000009.pdf

Other Examples

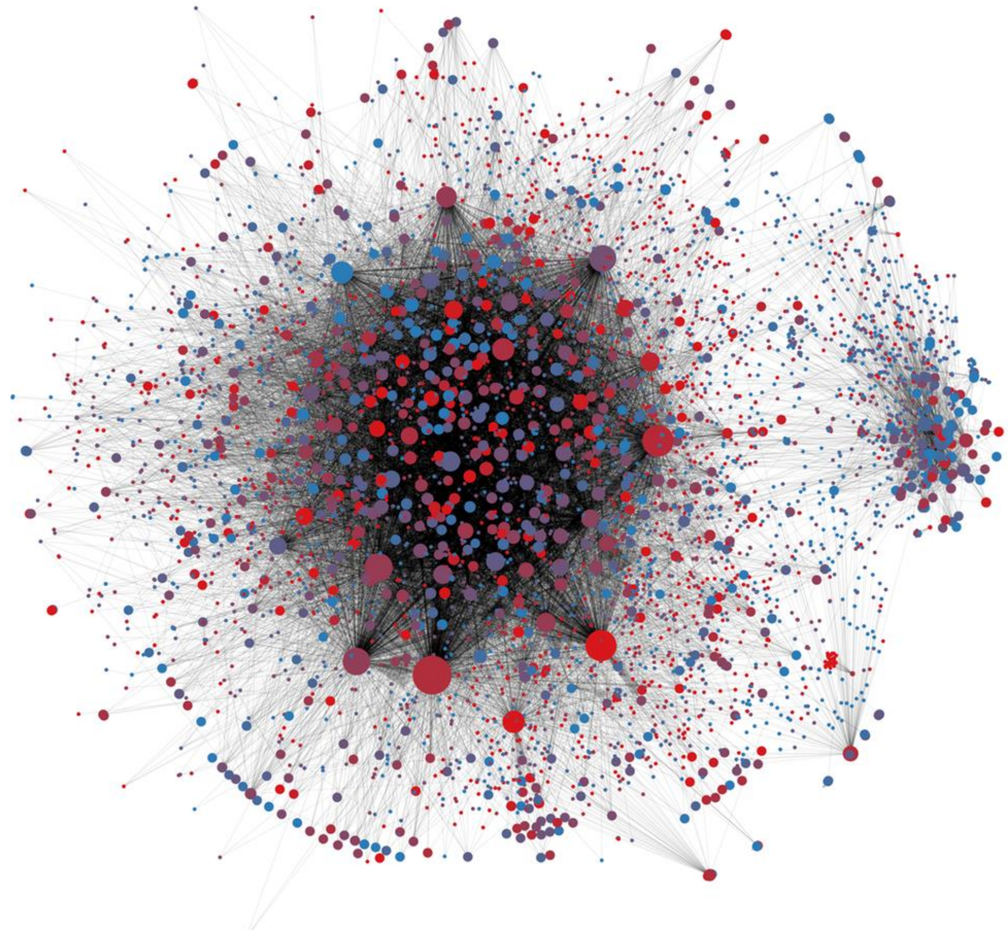
- Major League Baseball – Scheduling of games. Involves Matching and TSP



- Google search results.
- Social Networks – Finding influential people and communities
- Determining Aircraft routes, schedules, crew rotation, maintenance

Influence in Social Networks

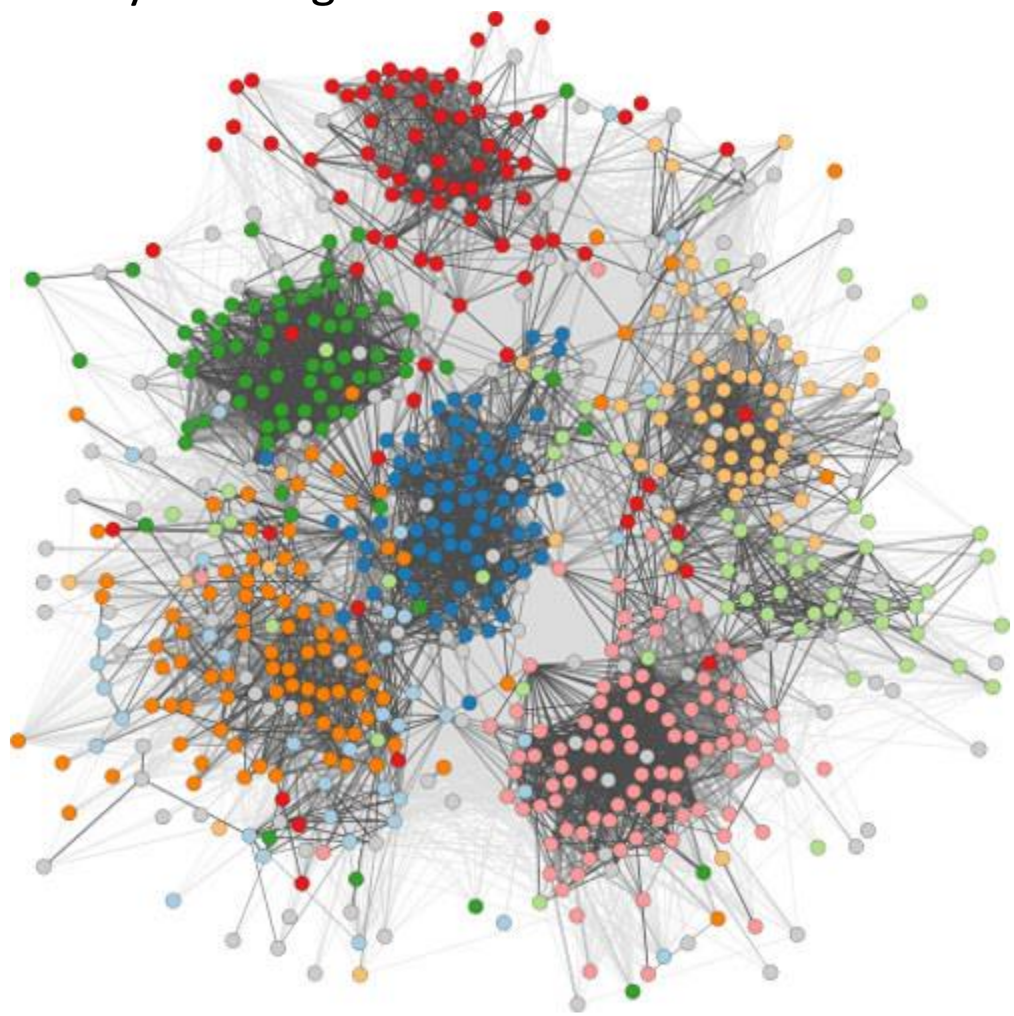
- Influential people are important in a social network. Facebook can charge a premium by targeting ads to those people.



http://theconversation.com/misinformation-on-social-media-can-technology-save-us-69264

Cliques in Social Networks

- A *clique* is a set of nodes that share certain characteristics. These are also important to marketers so that they can target ads.



https://lostcircles.com/