Aalto University
School of Science

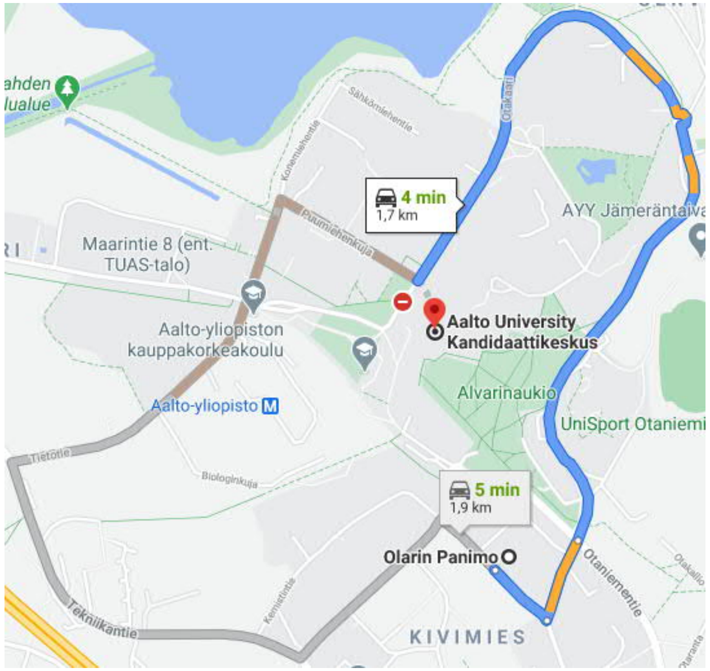
The shortest path problem

Emil Nyman
Presentation *10*
16.10.2020

MS-E2191 Graduate Seminar on Operations Research
Fall 2020

Motivation

What is the fastest route from point A to point B?



Content

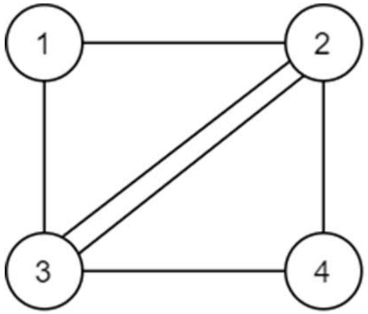
1. Background
2. Problem formulation
3. Generic algorithm
4. Implementations of the general algorithm
5. Applications

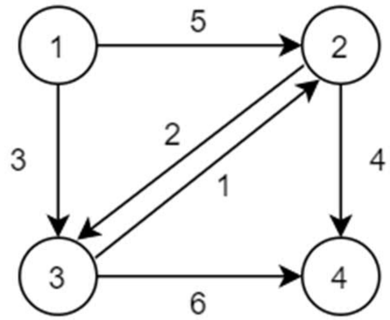
Background

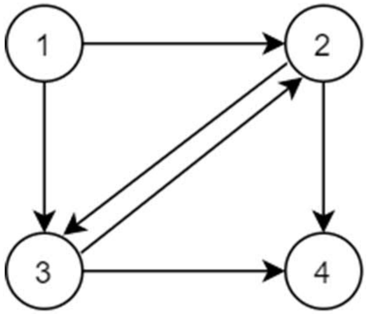
Aalto University
School of Science

Graph theory

- **Mathematical structures modeling relations between objects**
- **A graph consists of nodes and arcs**
- **Can be undirected or directed**
- **Can be cyclic or acyclic**
- **Arcs can have costs**







Problem formulation

Aalto University
School of Science

Shortest path problem

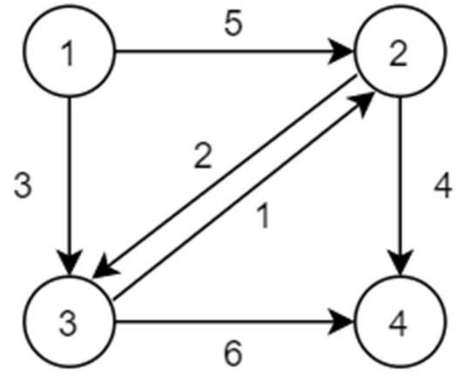
- We are given a directed graph (N, A) with nodes numbered $1, \dots, N$ and arcs $(i,\ j) \in A$ with a cost a_{ij}. The cost of a forward path $(i_1, i_2, \dots, i_k)$ is the cost of it arcs

$$\sum_{n=1}^{k-1} a_{i_n i_{n+1}}\,.$$

- This path is the shortest if it has a minimum length over all forward paths with the same origin and destination nodes

Example

- **What is the shortest path from node 1 to node 4?**
- **Trial and error**
- **Easy to solve**



Larger problems?

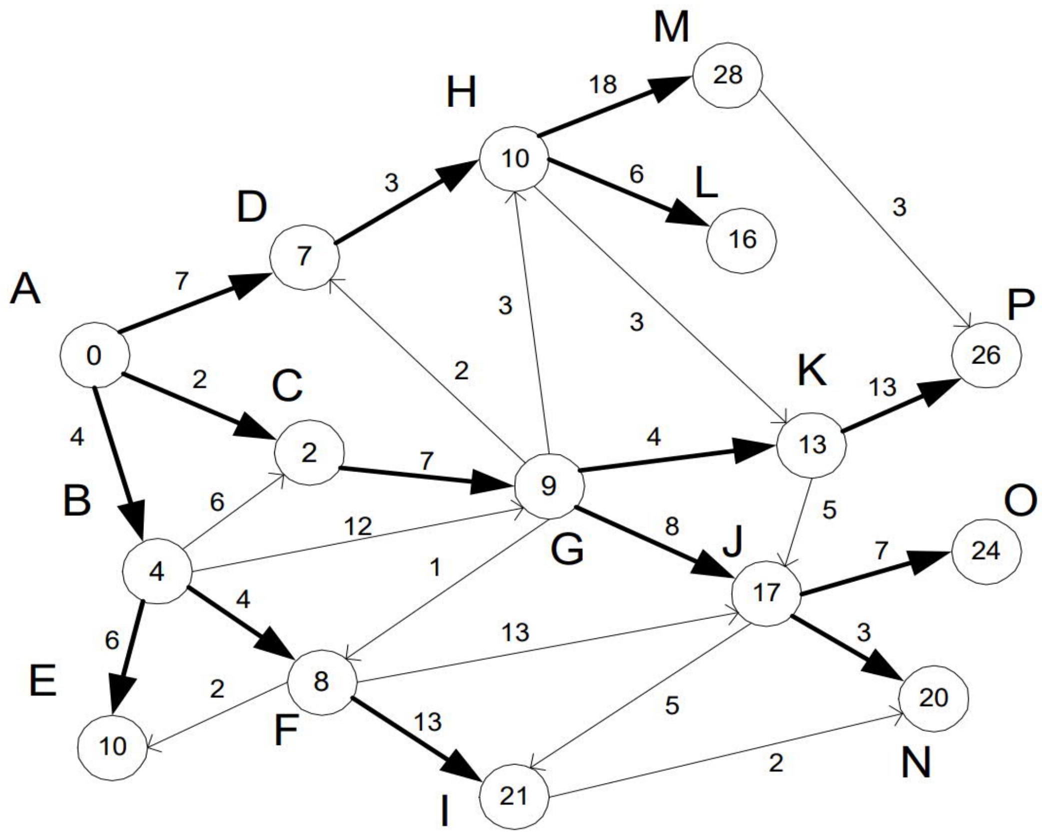


IMAGE: Xiao, Bin, Qingfeng ZhuGe, and EH-M. Sha. "Minimum dynamic update for shortest path tree construction." *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*. Vol. 1. IEEE, 2001.

Generic algorithm

Aalto University
School of Science

Generic shortest path algorithm 1/4

- The algorithm maintains a label d_i for each node $i \in N$ that is assumed to be a non-negative scalar that represents the cost of some path P from the start node to i
- In the beginning labels $d_i = \infty \; \forall \, i \neq 0$ and $d_0 = 0$
- Let the labels satisfy the complementary slackness (CS) condition for the shortest path problem

$$d_j \leq d_i + a_{ij} \quad \forall \, (i, j) \in A.$$

If

$$d_j = d_i + a_{ij} \quad \forall \, (i, j) \in P$$

then P is the shortest path from start node to i.

Generic shortest path algorithm 2/4

- **The algorithm goes through every arc to check violations of the CS condition**
- **If arc (i, j) violates the CS condition i.e. $d_j > d_i + a_{ij}$, then set $d_j := d_i + a_{ij}$ until the CS condition is satisfied for all arcs**
- **For efficiency we maintain a list of nodes V, called candidate list from which the algorithm chooses one node at a time and checks for CS conditions**

Generic shortest path algorithm 3/4

- **One iteration of the generic algorithm:**

  Remove a node i from candidate list V. For each outgoing arc, if $d_j > d_i + a_{ij}$, set

  $$d_j := d_i + a_{ij}$$

  and add j to V if it does not already belong to V.

- **Continue iteration until V is empty**

Generic shortest path algorithm 4/4

- **Upon termination, all labels are equal to the corresponding shortest distances**

$$\boldsymbol{d_j} = \begin{cases} \min_{(\boldsymbol{i},\boldsymbol{j})\in \boldsymbol{A}} \{\boldsymbol{d_j} + \boldsymbol{a_{ij}}\}, & \forall\, \boldsymbol{j} \neq \boldsymbol{1} \\ \boldsymbol{0}, & \boldsymbol{j} = \boldsymbol{1} \end{cases},$$

and thus, the shortest path can be achieved by following arcs back from the final node so that

$$\boldsymbol{d_j} = \boldsymbol{d_i} + \boldsymbol{a_{ij}}.$$

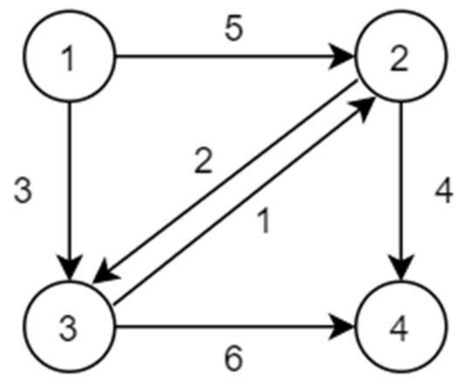
Example revised

- **Solving the shortest path problem from node 1 to node 4 using the generic algorithm**
- **One iteration of the generic algorithm:**

  Remove a node i from candidate list V. For each outgoing arc, if $d_j > d_i + a_{ij}$, set

$$d_j := d_i + a_{ij}$$

  and add j to V if it does not already belong to V.



Implementations of the general algorithm

Aalto University
School of Science

Implementations 1/3

- **The only difference in different implementations of the generic algorithm is how a node is removed from the candidate list V**
- **Can be categorized in two groups**
  - Label setting methods
  - Label correcting methods
- **Label setting methods have a better worse-case running time**
- **Label correcting methods are more flexible, better suited for advanced initialization and in practice have a faster running time**

Implementations: Label setting methods 2/3

- **Also called Dijkstra's algorithm**
- **Node removed from V is always the one with the minimum label**
- **Each node enters V at most once**
- **Implementations differ in obtaining the minimum at each iteration**
  - Binary heap
  - Dial's algorithm

Implementations: Label correcting method 3/3

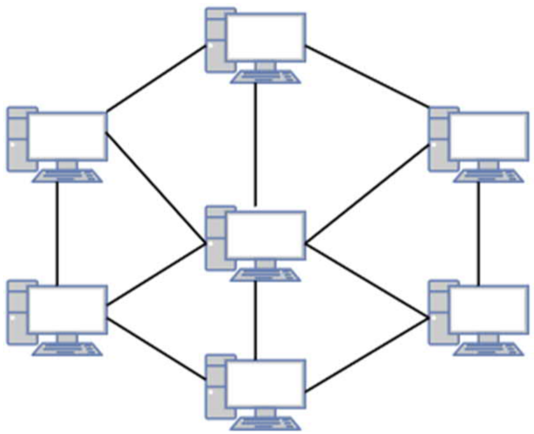
- **Node removed from V is done sorting the nodes in it**
- **Each node can enter V multiple times**
- **Label correcting methods**
  - Bellman-Ford: first-in first-out
  - D'Esopo-Pape: first-in first-out unless node has already been in V, then last-in first-out
  - SLF: if label is smaller than current smallest of V set new node first in queue, otherwise last
  - LLL: if first label is larger than the average of V, re-orgnaize V so that the first is last. Repeat until first node is smaller than average

Applications

Aalto University
School of Science

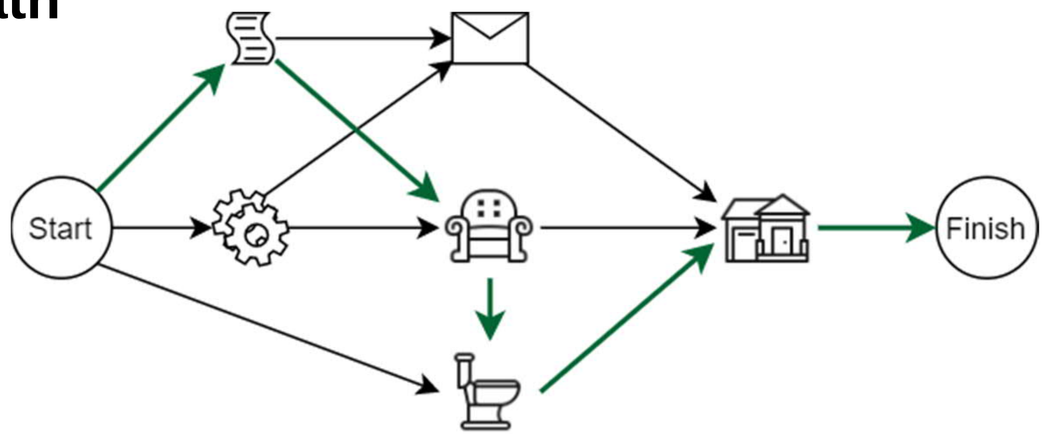
Applications: Data Networks

- Nodes represent computers
- Arcs represent commucation links
- Costs can represent: length, transmission capacity, traffic etc.
- Costs can differ over time



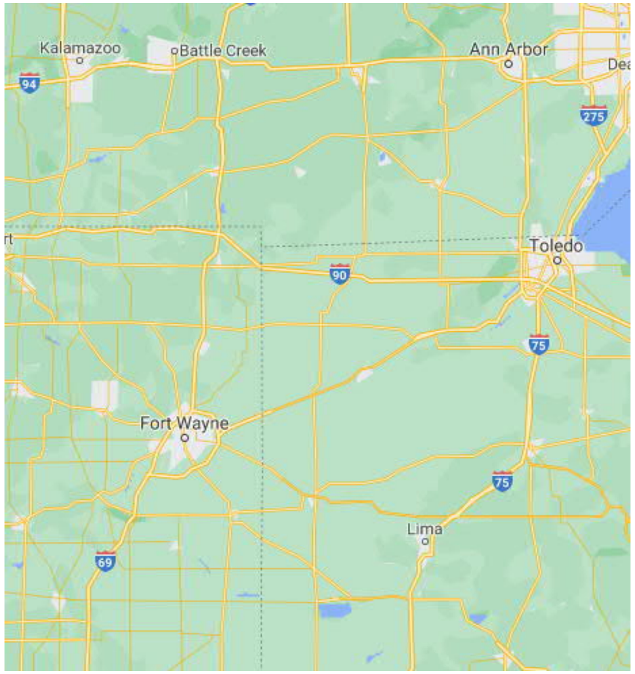
Applications: Project management

- **Finding the longest path from start to finish**
  - Can be viewed as a shortest path problem with negative cost
  - Must be acyclic
- **Critical path**



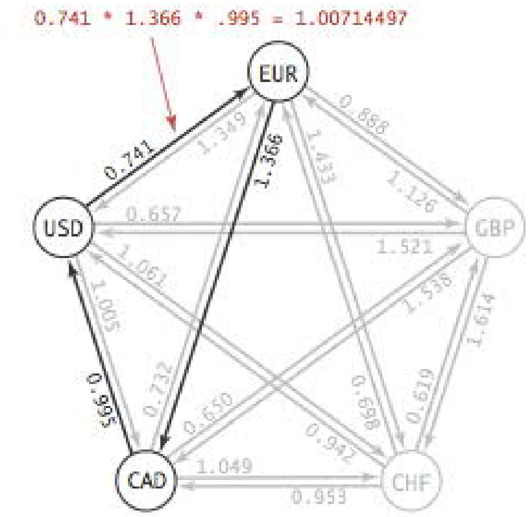
Applications: Road networks

- **Nodes represent junctions**
- **Arcs represent the roads**
- **Costs represent length, travel time etc.**
- **One-way roads with directional arcs**



Applications: Arbitrage detection

- **Arbitrage = making money without risk**
- **Finding arbitrages by detecting negative cycles**
  - E.g. Bellman-Ford method if the algorithm continues over N cycles

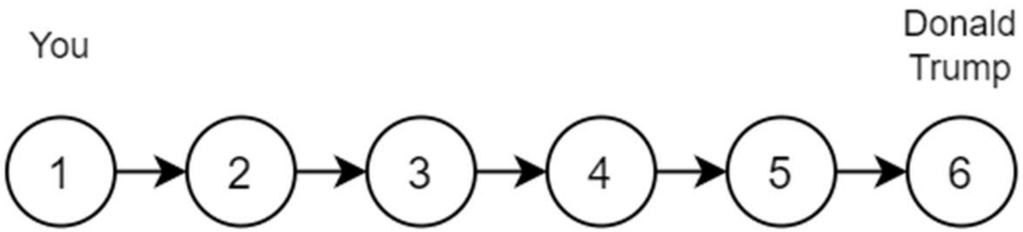


An arbitrage opportunity

IMAGE: https://algs4.cs.princeton.edu/44sp/

Applications: Social networks

- **Six degrees of separation**



Thank you!

Aalto University
School of Science

References

Content based on:

Bertsekas, Dimitri P. *Network optimization: continuous and discrete models*. Belmont, MA: Athena Scientific, 1998.

Larger network problem image:

Xiao, Bin, Qingfeng ZhuGe, and EH-M. Sha. "Minimum dynamic update for shortest path tree construction." *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*. Vol. 1. IEEE, 2001.

Map image:

https://www.google.fi/maps

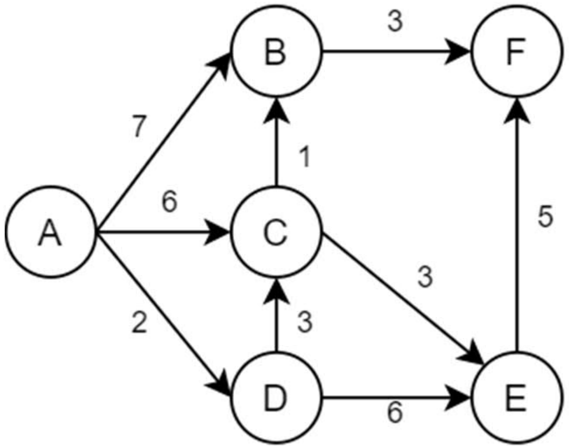
Arbitrage image:

https://algs4.cs.princeton.edu/44sp/

Homework: Dijkstra's shortest path algorithm

Given the following graph, where arcs connecting nodes A to F have different costs, find the shortest path from node A to node F by using Dijkstra's algorithm.

Report the labels d_i and the candidate list V in each iteration. What path is the shortest based on Dijkstra's algorithm?



DL 23.10. answers to: emil.nyman@aalto.fi