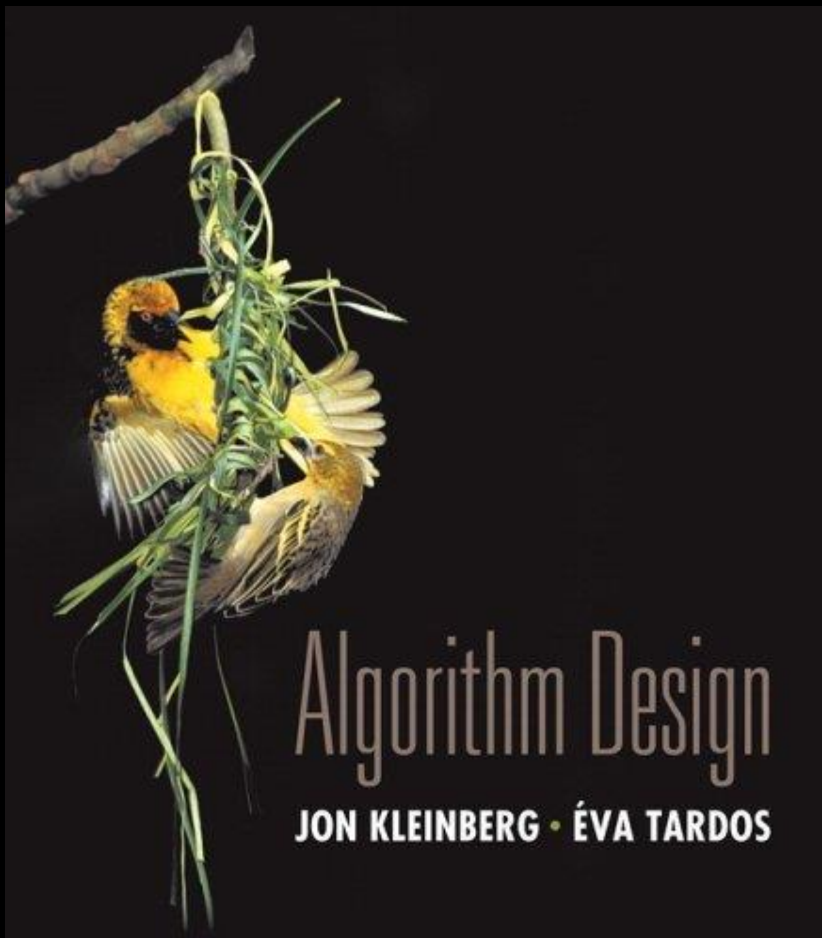


Chapter 6

Dynamic Programming

Slides by Kevin Wayne.
Copyright © 2005 Pearson-Addison Wesley.
All rights reserved.

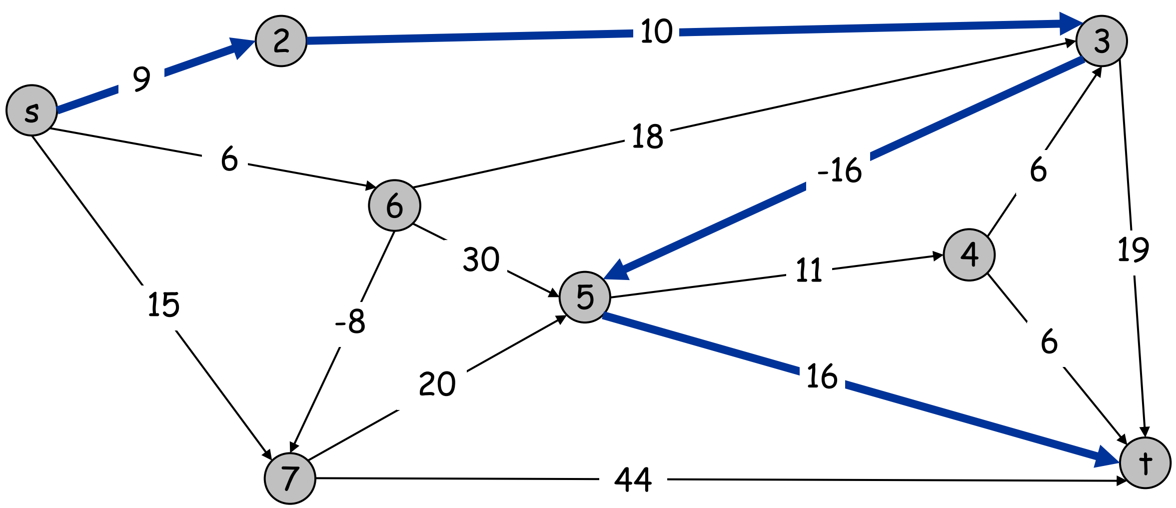
6.8 Shortest Paths

Shortest Paths

Shortest path problem. Given a directed graph $G = (V, E)$, with edge weights c_{vw}, find shortest path from node s to node t.

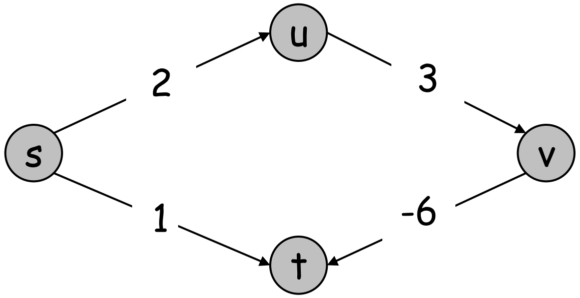
allow negative weights

Ex. Nodes represent agents in a financial setting and c_{vw} is cost of transaction in which we buy from agent v and sell immediately to w.

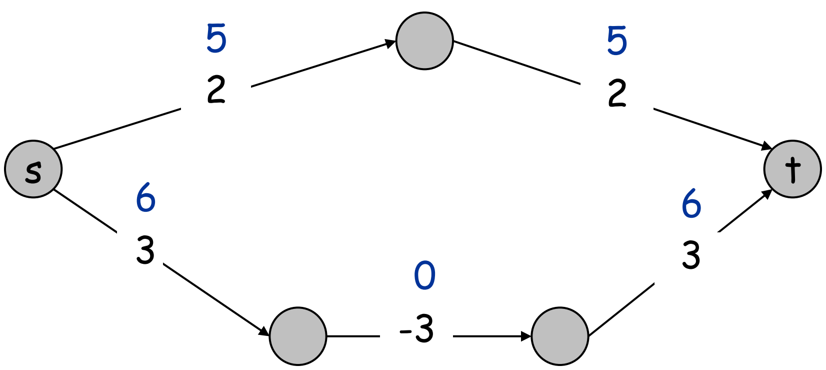


Shortest Paths: Failed Attempts

Dijkstra. Can fail if negative edge costs.

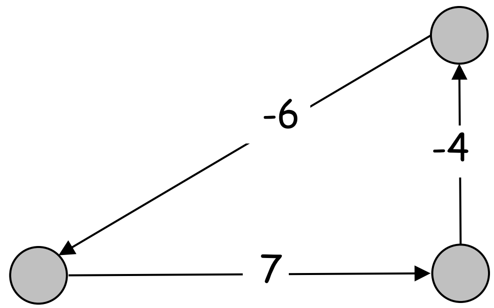


Re-weighting. Adding a constant to every edge weight can fail.



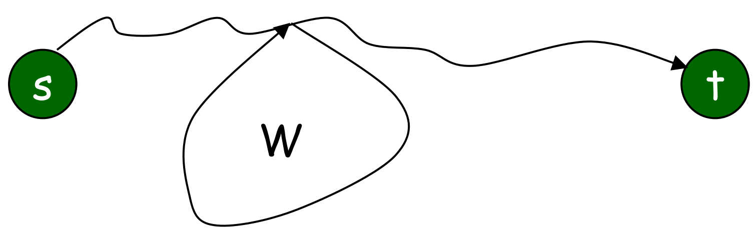
Shortest Paths: Negative Cost Cycles

Negative cost cycle.



Observation.

- If some path from s to t contains a negative cost cycle, there does not exist a shortest s-t path
- Otherwise, there exists one that is simple (no cycles and at most n-1 edges).



$c(W) < 0$

Shortest Paths: Dynamic Programming

Def. OPT(i, v) = length of shortest v-t path P using at most i edges.

- If (v, w) is first edge, then OPT uses (v, w), and then selects best w-t path using at most i-1 edges

$$OPT(i,v) = \begin{cases} 0, \text{if } v = t \\ \min_{(v,w)\in E}\left\{c_{vw} + OPT(i-1,w)\right\} & \text{otherwise} \end{cases}$$

Remark. By previous observation, if no negative cycles, then OPT(n-1, v) = length of shortest v-t path.

Shortest Paths: Implementation

```
Shortest-Path(G, t) {
   foreach node v ∈ V
      M[0, v] ← ∞

   for i = 1 to n-1
      M[i, t] ← 0

   for i = 1 to n-1
      foreach node v ∈ V
         M[i, v] ← ∞

         foreach edge (v, w) ∈ E
                if M[i, v]> M[i-1, w] + c_vw
                    M[i,v] ← M[i-1, w] + c_vw
                    sucessor[v] ← w
}
```

Analysis. $\Theta(mn)$ time, $\Theta(n^2)$ space.

Finding the shortest paths. Maintain a "successor" for each table entry.

6.10 Negative Cycles in a Graph

Detecting Negative Cycles

Lemma. If OPT(n,v) < OPT(n-1,v) for some node v, then there is a negative cycle in the graph that includes a node that reaches t .

- If you gain by using n edges => gain from using non-simple path => negative cycle

Lemma. If OPT(n,v) = OPT(n-1,v) for all v, then no negative cycles that includes a node that reaches t.

Detecting Negative Cycles: Summary

Detecting negative cycles:

- Run Bellman-Ford for n iterations (instead of n-1).
- If there is a node v with $M[n, v] < M[n-1, v]$ => there is neg cycle
- Otherwise, there is no negative cycle

To find the negative cycle:

- Take vertex v with $M[n, v] < M[n-1, v]$, look at its shortest path to t
  => must contain a negative cycle