Shortest Paths

How to find the shortest route between two points in a map

INPUT:
- A directed graph G=(V,E)
- A weight function w:E → R

Weight of path p=<v_0, v_1, ..., v_k> is
w(p)=\sum_{i=1}^{k} w(v_{i-1}, v_i)
Weight of a Shortest Path

Shortest-path weight from u to v
- \( d(u,v) = \min \{ w(p) : p = u \rightarrow v \} \)
  if there is a path from u to v
- \( d(u,v) = \infty \) otherwise

Shortest-path from u to v
- Any path \( p = u \rightarrow v \) with weight
  \( w(p) = d(u,v) \)
The Weight Function

- Can think of weights as something that:
  - Accumulate linearly along the path
  - We want to minimize

- Examples:
  - Times, costs, penalties, losses ...

- Generalization of Breadth-First Search to weighted graphs
Variants, 1

- **Single-Source Shortest Paths**
  - Find a shortest path from a given source vertex s to every other vertices v

- **Single-Destination Shortest Paths**
  - Find a shortest path to a given destination vertex t from every other vertices v
Variants, 2

- **Single-Pair Shortest Paths**
  - Find a shortest path from a given vertex $u$ and a given vertex $v$

- **All-Pair Shortest-Paths**
  - Find a shortest path from vertex $u$ to vertex $v$ for every pair of vertices $u$ and $v$
Negative-weight Edges

- Ok, as long as no negative cycles are reachable from the source.

- Negative cycle → we keep going around it obtaining $w(s \rightarrow v) = -\infty$ for each $v$ in the cycle.

- Some algorithm do not tolerate negative-weight edges at all.
Optimal Substructure

Lemma: Every sub-path of a shortest path is a shortest path

Proof: Assume \( p = u \rightarrow v \) is a shortest path such that \( p = u \rightarrow x \rightarrow y \rightarrow v \) and \( w(p) = w(u \rightarrow x) + w(x \rightarrow y) + w(y \rightarrow v) \).

Now suppose that \( x \rightarrow y \) is a path shorter than \( x \rightarrow y \). Hence, \( w(x \rightarrow y) < w(x \rightarrow y) \). But then \( p' = u \rightarrow x \rightarrow y \rightarrow v \) is shorter than \( p \). A contradiction.
Cycles

- Shortest paths cannot contain cycles
  - Negative-weight cycles are already ruled out
  - Positive-weight $\Rightarrow$ we can obtain a shortest path by omitting the cycle
  - Zero-weight. No reason to use them (this we will assume)
For $v \in V$ the output is $d[v] = d(s,v)$
- Initially $d[v] = \infty$
- Reduces as algorithm progresses, but always $d[v] \geq d(s,v)$
- Call $d[v]$ a **shortest path estimate**

$p[v] =$ the predecessor of $v$ in a path to $s$
- If no predecessor, $p[v] = \text{NIL}$
- $p$ induces a tree—**Shortest-path tree**
Initialization

All shortest-paths algorithms start with

Init-Single-Source(V,s)
for each $v \in V$ do
  $d[v] = \infty$
  $\pi[v] = \text{NIL}$
  $d[s] = 0$
Relaxation

Can we improve the shortest-path estimated for v going through u and taking (u,v)?

\[ \text{Relax}(u,v,w) \]
if \( d[v] > d[u] + w(u,v) \)
then \( d[v] = d[u] + w(u,v) \)
\( \pi[v] = u \)
Scheme for Single-Source Shortest-Paths Algorithms

- Start by calling Init-Single-Source
- Relax edges
- Different algorithms differ on
  - Number of relaxations
  - Order of relaxations
- Bellman-Ford
- Dijkstra
Shortest-Path Properties

Based on calling Init-Single-Source once and Relax zero or more times

Lemma: Triangle inequality

For all \((u,v) \in E\): \(d(s,v) \leq d(s,u) + w(u,v)\)

Proof: Weight of shortest path \(s \rightarrow v\) is \(\leq\) weight of any path \(s \rightarrow v\). Path \(s \rightarrow u \rightarrow v\) is a path from \(s\) to \(v\), and if \(s \rightarrow u\) is a shortest path its weigh is \(d(s,u) + w(u,v)\)
Upper-bound Property

Lemma: Always have $d[v] \geq \delta(s,v)$ for all $v$. When $d[v] = \delta(s,v)$ it never changes.

Proof: Initially true. Suppose $v$ such that $d[v] < \delta(s,v)$, and wlog $v$ is the first vertex for which this happens. Let $u$ the vertex that updates $d[v]$ to $d[u] + w(u,v)$. So ...
Upper-bound Property, 2

\[ d[v] < d(s,v) \]
\[ \leq d(s,u) + w(u,v) \text{ (triangle inequality)} \]
\[ \leq d[u] + w(u,v) \text{ (v is first violation)} \]
\[ \Rightarrow d[v] < d[u] + w(u,v) \text{ which contradicts } d[v] = d[u] + w(u,v). \]

Once \( d[v] \) reaches \( d(s,v) \), it never goes lower. It never goes up, since relaxations only lower estimates.
Other properties

- **No-path property**
  - If $d(s,v) = \infty$ then $d[v] = \infty$ always

- **Convergence property**
  - If $s \rightarrow u \rightarrow v$ is a shortest path, $d[u] = d'(s,u)$ and we call Relax$(u,v,w)$ then $d[v] = d(s,v)$ afterward

- **Path-relaxation property**
  - Let $p = <v_0, v_1, ... v_k>$ be a shortest path from $s=v_0$ to $v=v_k$. If we relax in order $(v_0,v_1), (v_1,v_2), ..., (v_{k-1},v_k)$ then $d[v_k] = d(s,v)$
Assignments

- Textbook, Chapter 24, pages 580—592
- Updated information on the class web page:
  
  www.ece.neu.edu/courses/eceg205/2004fa