Shortest Paths

Digraph with edge weights (costs, distances)

Shortest path from $s$ to $t$: path of minimum total wt.

Problems:

single pair: given $s, t$, find a shortest path from $s$ to $t$

single source: given $s$, find shortest paths from $s$ to all reachable vertices

all pairs: find shortest paths between all pairs

Cases:

acyclic

no negative wts

general

(planar, etc.)
Properties:

If a shortest path from $s$ to $t$ iff there is no negative (total wt.) cycle on a path from $s$ to $t$.

If there is no such cycle, there is a shortest path that is simple (no repeated vertex).

If no neg cycle reachable from $s$, then if shortest path tree rooted at $s$, contains all vertices reachable from $s$, all tree paths are shortest paths in graph.

New goal: find a negative cycle or construct a shortest path tree.

(single-source problem is central)
Given a spanning tree \(T\) rooted at \(s\),
\[ d(v) = \text{tree wt from } s \text{ to } v, \]
is \(T\) a shortest path tree?

Yes, iff there is no \((v,w)\) with \(d(v) + c(v,w) < d(w)\).

Edge relaxation algorithm to find a shortest path tree:
\[ d(s) = 0, \quad d(v) = \infty \text{ for } v \neq s \]

while exists edge \((v,w)\) with \(d(v) + c(v,w) < d(w)\)
\[ \text{do} \{ d(w) = d(v) + c(v,w); \quad p(w) = v \} \]
d\(v\) is always the wt of some \(s-v\) path

if algorithm stops and \(p\) defines a tree,
must be a shortest path tree

stops iff no neg cycle

(alg maintains \(d(w) \geq d(v) + c(v,w)\) if \(v = p(w)\))
Suppose \( T \) not a \( sp \) tree. Let \( x \) be such that \( d(x) > s-x \) distance. Let \( P \) be a shortest path from \( s \) to \( x \), \( d'(v) = P \)-distance from \( s \).

Let \( (v, w) \) be first edge along \( P \) such that \( d'(v) < d(w) \).

Then \( d(v) + c(v, w) = d'(v) + c(v, w) = d'(v) < d(w) \).

(This gives the hard direction of \( sp \) tree test.)

Suppose edge relaxation algorithm creates a cycle.

Then it must be a negative cycle.

\[
d(v) + c(v, w) < d(w) \Rightarrow d(v) - d(w) + c(v, w) < 0
\]

Sum around cycle: \[
\sum_{i=1}^{k} (d(v_i) - d(v_{i+1})) + c(v_{i+1}, v_i) < 0
\]
Labeling and scanning algorithm:

$L = \{s\}$; $d(s) = 0$; $d(v) = \infty$ for $v \neq s$;

while $L \neq \emptyset$ do \
    remove $v$ from $L$; \hspace{1cm} \text{scan}(v);$ for each $(v, w)$ do \
    if $d(v) + c(v, w) < d(w)$ then \
    \hspace{1cm} $d(w) = d(v) + c(v, w)$; $p(w) = v$; add $w$ to $L$ \}\n
\[\text{unlabeled}\]

\[\text{labeled}\]

\[\text{scanned}\]
Ayclic: topological $O(m)$

Non neg: shortest first (mind) (Dijkstra)
$O(n^2)$ original, $O(m \log n)$ standard heap

General: FIFO-scanning
queue = T

$O(nm)$

$O(nm \log C)$ (cost-scaling)
Dijkstra alg:

monotonicity on heap:
vis are deleted from l in increasing order by d:

$Dial: \text{small integer edge weights} \quad O(m + Cn)$

$\leq C$

$\frac{1}{C}$

$m + 2 \sqrt{Cn}$

$\frac{1}{C} (m + n \log C)$
All pairs:

Dynamic prog.

\[ P_2 \rightarrow P_3 \rightarrow y \]

\[ x \rightarrow P_1 \]

\[ d(x, x) = 0 \]

\[ d(x, y) = \infty \text{ for all } x \neq y, (x, y) \notin E \]

\[ d(x, y) = d(x, y) \text{ if } x \neq y, (x, y) \notin E \]

for \( z \)

for \( x \)

for \( y \)

if \( d(x, z) + d(z, y) < d(x, y) \) then

\[ d(x, y) = d(x, z) + d(z, y) \]

\[ O(n^3) \]
in single sources

\[ \Rightarrow \text{Dijkstra}: O(nm + n^2 \log n) \]

1 Bellman-Ford = eliminate negative edge costs

\[ p(v) \]

\[ c'(v, w) = c(v, w) + p(v) - p(w) \geq 0 \]

\[ s \xrightarrow{t} \]