Shortest-paths – Proofs

- Weight of path $p = v_0 \to v_1 \to \cdots \to v_k$:

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

- Shortest-path weight $u \rightsquigarrow v$

$$\delta(u,v) = \begin{cases} \min\{w(p) : u \overset{p}{\rightsquigarrow} v\} & \text{if there exists a path } u \rightsquigarrow v \\ \infty & \text{otherwise} \end{cases}$$

- Shortest-path $u \rightsquigarrow v$

any path p such that $w(p) = \delta(u,v)$

Shortest-paths – Proofs

Triangular inequality:

for all $(u, v) \in E$, $\delta(u, v) \leq \delta(u, x) + \delta(x, v)$.

Proof: Note that

$$\underline{\text{Weight of shortest path } s \rightsquigarrow v} \leq \underline{\text{weight of any path } s \rightsquigarrow v}$$

The path $s \rightsquigarrow u \rightarrow v$ *is a path* $s \rightsquigarrow v$*, and if we use a shortest path* $s \rightsquigarrow u$*, its weight is* $\delta(u, x) + \delta(x, v)$*.*

Shortest-paths – Proofs

Upper-bound property:

> *Always have* $d[v] \geq \delta(s, v)$ *for all* v.
> *Once* $d[v] = \delta(s, v)$, *it never changes.*

Proof. Initially true. Suppose there exists a vertex such that $d[v] < \delta(s, v)$. Without loss of generality, v is first vertex for which this happens. Let u be the vertex that causes $d[v]$ change. Then $d[v] = d[u] + w(u, v)$. So

$$\begin{aligned} d[v] &< \delta(s, v) \\ &\leq \delta(s, u) + w(u, v) \\ &\leq d[u] + w(u, v) \end{aligned}$$

which implies $d[v] < d[u] + w(u, v)$. Contradicts $d[v] = d[u] + w(u, v)$. Once $d[v]$ reaches $\delta(s, v)$, it never goes lower. It never goes up, since relaxations only lower shortest-path weights.

Shortest-paths – Proofs

No-path property:

If $\delta(s, v) = \infty$, then $d[v] = \infty$ always.

Proof. $d[v] \geq \delta(s, v) = \infty$ implies that $d[v] = \infty$.

Shortest-paths – Proofs

Convergence property:

If $s \leadsto u \to v$ is a shortest-path, and $d[u] = \delta(s,u)$. Then after "Relax $u \to v$", $d[v] = \delta(s,v)$.

Proof. After relaxation

$$\begin{aligned} d[v] &\leq d[u] + w(u,v) \\ &= \delta(s,u) + w(u,v) \\ &= \delta(s,v) \end{aligned}$$

On the other hand, we have $d[v] \geq \delta(s,v)$. Therefore, it must have $d[v] = \delta(s,v)$.

Shortest-paths – Proofs

Path relaxation property

Let $p = v_0 \to v_1 \to \cdots \to v_k$ *be a shortest-path. If we relax in order,* $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$*, even intermixed with other relaxations, then* $d[v_k] = \delta(v_0, v_k)$*.*

Proof. Induction to show $d[v_i] = \delta(s, v_i)$ after (v_{i-1}, v_i) is relaxed.

- Basis step: $i = 0$. Initially $d[v_0] = \delta(s, v_0) = \delta(s, s)$
- Inductive step: Assume $d[v_{i-1}] = \delta(s, v_{i-1})$. Relax (v_{i-1}, v_i). By convergence property, $d[v_i] = \delta(s, v_i)$ afterward and $d[v_i]$ never changes.

Shortest-paths – Proofs

Correctness of the Bellman-Ford algorithm

It is guaranteed to converge after $|V| - 1$ passes, assuming no negative-weight cycles.

Proof. Use path-relaxation property.
Let v be reachable from s, and let $p = v_0 \to v_1 \to \cdots \to v_k$ be the shortest path from s to v, where $v_0 = s$ and $v_k = v$.
Since p is acyclic, it has $\leq |V| - 1$ edges, so that $k \leq |V| - 1$ edges.
Each iteration of the **for** loop realxes all edges:

- First iteration relaxes (v_0, v_1)
- Second iteration relaxes (v_1, v_2)
- ...
- kth iteration relaxes (v_{k-1}, v_k)

By the path-relaxation property, $d[v] = d[v_k] = \delta(s, v_k) = \delta(s, v)$.

Shortest-paths – Proofs

Correctness of Dijkstra's algorithm

Show that $d[u] = \delta(s, u)$ when u is added to S in each iteration.

Proof:

- We prove by contradiction. Suppose there exists u such that $d[u] \neq \delta(s, u)$. Without loss of generality, let u be the first vertex for which $d[u] \neq \delta(s, u)$ when u is added to S in each iteration.
- Observation:
  - $u \neq s$, since $d[s] = \delta(s, s) = 0$.
  - Therefore, $s \in S$ and $S \neq \emptyset$
  - There must have be some path $s \leadsto u$, since otherwise $d[u] = \delta(s, u) = \infty$ by no-path property.

  So, there is a path $s \leadsto u$. Then there is a shortest path $s \overset{p}{\leadsto} u$.
- Just before u is added to S, path p connects a vertex in S (i.e., s) to a vertex in $V - S$ (i.e., u). Let y be first vertex along p that's in $V - S$ and and let x be y's predecessor.
- Decompose p into

$$s \overset{p_1}{\leadsto} x \rightarrow y \overset{p_2}{\leadsto} u$$

  (could have $x = s$ or $y = u$, so that p_1 or p_2 may have no edges.)

Shortest-paths – Proofs

Correctness of Dijkstra's algorithm, cont'd

- Claim:[1] $d[y] = \delta(s, y)$ when u is added to S.
- Now we can get a contradiction to $d[u] \neq \delta(s, u)$:

  y is on shortest path $s \leadsto u$, and all edge weights are nonnegative

  $$\Downarrow$$
  $$\delta(s, y) \leq \delta(s, u)$$
  $$\Downarrow$$
  $$d[y] = \delta(s, y) \leq \delta(s, u) \leq d[u] \text{ (upper bound property)}$$

  Also, both y and u were in Q when we chose u, so that

  $$d[u] \leq d[y]$$

  Therefore, $d[y] = \delta(s, y) = \delta(s, u) = d[u]$. Contradicts assumption that $d[u] \neq \delta(s, u)$.
- Hence, Dijkstra's algorithm is correct.

[1] Proof. $x \in S$ and u is the first vertex such that $d[u] = \delta(s, u)$ when u is added to S $\Rightarrow d[x] = \delta(s, x)$ when x is added to S. Relaxed (x, y) at that time, so by the convergence property, $d[y] = \delta(s, y)$.