Shortest Path Algorithms - continued

Navigation from source A to node B
→ mapped to a problem in graph
\[
G = (V, E) \quad \text{directed and weighted}
\]
\[
W: E \to \mathbb{R}
\]

Dijkstra algorithm: Single source shortest path (SSSP)
\[
W: E \to \mathbb{R}^+
\]
(No -ve weight)

Bellman-Ford: SSSP

Directed Acyclic Graph (DAG) no -ve cycles

Inspite of -ve cycles, can we demand a shortest path that has no cycles

Simple path: \( V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow \cdots \rightarrow V_k \)
How many paths between S and D?

Brute force $\Rightarrow$ exponential time

Common step between BF and D

Short cut step

$S(s) = 0$

$S(v) \leq \min \{ S(u_1) + w_1, S(u_2) + w_2, S(u_3) + w_3 \}$

$S(v)$: upper bound

Gradually, these algorithms refine the estimate.

Induction invariant: There is a path from s to v with distance $\leq S(v)$.
Shortcut step (Relaxation Step)
Relax edge \((u, v)\)

if \(s(v) > s(u) + w(u, v)\)
- then \(s(v) \leftarrow s(u) + w(u, v)\)

In BF, we have \(|V| - 1\) iteration
where in each iteration, we
relax all edge \((u, v) \in E\)
(ordering doesn't matter)

Running time: \(O(|V||E|) \sim O(n^3)\)

\(|V| = n\) \(|E| = m\)

Correctness: - No path can have more
- than \(|V| - 1 \) edges

Claim: All vertices \(V\) whose shortest path
from the source \(S\) consists of \(k\) links
will have \(s(v) = D(v)\) with \(k\) iterations

shortest path
lengths
Proof by induction (on \( k \))

**Basis:** \( k = 0 \): by initialization, it is correct

**Suppose true for** \([0, 1, 2, \ldots, i-1]\) \( \Rightarrow \)

\([i-1]\)

In the \( i \)th iteration, when we relax edge \((u, v)\)

\( D(v) = D(v)\)