Single-Source Shortest Paths

Problem: To find the shortest path from point A to point B on a map. You are given a map with distances between adjacent nodes already marked.

Solution using brute force: Enumerate the total distance using all paths (eliminating cycles), and select the shortest. Leads to a combinatorial explosion of possibilities.

Shortest-path problem

- Given a weighted directed graph $G = (V, E)$
- Weight function $w = E \to \mathbb{R}$ to map edges to real-valued weights
- Weight of path $p = \langle v_0, v_1, \ldots, v_k \rangle$
  \[
  w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)
  \]
- Shortest path weight from $u$ to $v$
  \[
  \delta(u, v) = \begin{cases} 
  \min\{w(p) : u \xrightarrow{p} v\} & \text{if there is a path from } u \text{ to } v \\
  \infty & \text{otherwise}
  \end{cases}
  \]
- A shortest path from vertex $u$ to vertex $v$ is defined as any path $p$ with weight $w(p) = \delta(u, v)$.
  - Breadth-first search algorithm is an example of shortest path algorithm that works on unweighted graphs
    - Each edge is considered to be of unit weight
- Representing shortest paths
  - For each vertex $v \in V$, maintain a predecessor $\pi[v]$
  - Find predecessor subgraph $G_\pi = (V_\pi, E_\pi)$ induced by $\pi$ values
  - A shortest-paths tree rooted at $s$ is a directed subgraph $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$, such that
    1. $V'$ is the set of vertices reachable from $s$ in $G$
    2. $G'$ forms a rooted tree with $s$
    3. For all $v \in V'$, the unique simple path from $s$ to $v$ in $G'$ is a shortest path from $s$ to $v$ in $G$

Shortest Paths and Relaxation

- Repeatedly decrease an upper bound on the actual shortest-path weight of each vertex until the upper bound equals the shortest-path weight
- Optimal substructure of a shortest path
  - A shortest path between two vertices contains other shortest paths within it

Lemma 1 (Subpaths of shortest paths are shortest paths.) Given a weighted, directed graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$, let $p = \langle v_1, v_2, \ldots, v_k \rangle$ be a shortest path from vertex $v_1$ to vertex $v_k$ and, for any $i$ and $j$ such that $1 \leq i \leq j \leq k$, let $p_{ij} = \langle v_i, v_{i+1}, \ldots, v_j \rangle$ be the subpath of $p$ from vertex $v_i$ to vertex $v_j$. Then, $p_{ij}$ is a shortest path from $v_i$ to $v_j$.

Corollary 1 Let $G = (V, E)$ be a weighted, directed graph with weight function $w : E \to \mathbb{R}$. Suppose that a shortest path $p$ from a source $s$ to a vertex $v$ can be decomposed into $s \xrightarrow{p'} u \to v$ for some vertex $u$ and path $p'$. Then, the weight of a shortest path from $s$ to $v$ is $\delta(s, v) = \delta(s, u) + w(u, v)$.

Lemma 2 Let $G = (V, E)$ be a weighted, directed graph with weight function $w : E \to \mathbb{R}$ and source vertex $s$. Then, for all edges $(u, v) \in E$, we have $\delta(s, v) \leq \delta(s, u) + w(u, v)$.
Relaxation

- For each vertex $v \in V$, maintain an attribute $d[v]
- \quad d[v]$ – upper bound on the weight of a shortest path from source $s$ to $v$ – shortest path estimate
- Initialization procedure
  
  \begin{verbatim}
  initialize_single_source (G,s)
  for each vertex $v \in V[G]$ do
    $d[v] \leftarrow \infty$
    $\pi[v] \leftarrow \text{nil}$
    $d[s] \leftarrow 0$
  
  relaxation
  \end{verbatim}


Properties of relaxation

Lemma 3 Let $G = (V,E)$ be a weighted, directed graph with weight function $w : E \rightarrow \mathbb{R}$, and let $(u,v) \in E$. Then, immediately after relaxing edge $(u,v)$ by executing $\text{relax}(u,v,w)$, we have $d[v] \leq d[u] + w(u,v)$.

Lemma 4 Let $G = (V,E)$ be a weighted, directed graph with weight function $w : E \rightarrow \mathbb{R}$. Let $s \in V$ be the source vertex, and let the graph be initialized by $\text{initialize_single_source}(G,s)$. Then, $d[v] \geq \delta(s,v)$ for all $v \in V$, and this invariant is maintained over any sequence of relaxation steps on the edges of $G$. Moreover, once $d[v]$ achieves its lower bound $\delta(s,v)$, it never changes.

Corollary 2 Suppose that in a weighted, directed graph $G = (V,E)$ with weight function $w : E \rightarrow \mathbb{R}$, no path connects a source vertex $s \in V$ to a given vertex $v \in V$. Then, after the graph is initialized by $\text{initialize_single_source}(G,s)$, we have $d[v] = \delta(s,v)$, and this equality is maintained as an invariant over any sequence of relaxation steps on the edges of $G$.

Lemma 5 Let $G = (V,E)$ be a weighted, directed graph with weight function $w : E \rightarrow \mathbb{R}$, let $s \in V$ be a source vertex, and let $s \leadsto u \rightarrow v$ be a shortest path in $G$ for some vertices $u,v \in V$. Suppose that $G$ is initialized by $\text{initialize_single_source}(G,s)$ and then a sequence of relaxation steps that includes the call $\text{relax}(u,v,w)$ is executed on the edges of $G$. If $d[u] = \delta(s,u)$ at any time prior to the call, the $d[v] = \delta(s,v)$ at all times after the call.

Shortest-paths trees

Lemma 6 Let $G = (V,E)$ be a weighted, directed graph with weight function $w : E \rightarrow \mathbb{R}$ and source vertex $s \in V$, and assume that $G$ contains no negative-weight cycles that are reachable from $s$. Then, after the graph is initialized by $\text{initialize_single_source}(G,s)$, the predecessor subgraph $G_{\pi}$ forms a rooted tree with root $s$, and any sequence of relaxation steps on edges of $G$ maintain this property as an invariant.

Lemma 7 Let $G = (V,E)$ be a weighted, directed graph with weight function $w : E \rightarrow \mathbb{R}$ and source vertex $s \in V$, and assume that $G$ contains no negative-weight cycles that are reachable from $s$. Let us call \text{initialize_single_source}(G,s) and then execute any sequence of relaxation steps on edges of $G$ that produces $d[v] = \delta(s,v)$ for all $v \in V$. Then, the predecessor subgraph $G_{\pi}$ is a shortest-paths tree rooted at $s$.

Dijkstra's Algorithm

- All edge weights are assumed to be non-negative
- Maintain a set \( S \) of vertices whose final shortest-path weights from the source \( s \) have already been determined, or for all vertices \( v \in S \), we have \( d[v] = \delta(s,v) \).
- Select the vertex \( u \in V - S \) with the minimum shortest-path estimate, using a priority queue \( Q \)
- Insert \( u \) into \( S \)
- Relax all edges leaving \( u \)

**Dijkstra (G,w,s)**

```plaintext
initialize_single_source(G, s)
S ← ∅
Q ← V[G]
while Q ≠ ∅ do
    u ← extract_min(Q)
    S ← S ∪ \{u\}
    for each vertex \( v \in Adj[u] \) do
        relax(u, v, w)
```

**Analysis**
- Each \( extract\_min \) – \( O(V) \)
- Total time for \( extract\_min \) – \( O(V^2) \)
- \( |E| \) iterations of for loop with \( O(1) \) for each iteration
- Total run time – \( O(V^2 + E) = O(V^2) \)

**Bellman-Ford Algorithm**

```plaintext
bellman_ford (G,w,s)
initialize_single_source (G, s)
for i ← 1 to |V[G]| - 1 do
    for each edge \( (u,v) \in E[G] \) do
        relax (u, v, w)
    for each edge \( (u,v) \in E[G] \) do
        if \( d[v] > d[u] + w(u,v) \) then
            return false
    return true
```

**Lemma 8** Let \( G = (V, E) \) be a weighted, directed graph with source \( s \) and weight function \( w : E \to \mathbb{R} \), and assume that \( G \) contains no negative weight cycles that are reachable from \( s \). Then, at the termination of \( \text{BELLMAN-FORD} \), we have \( d[v] = \delta(s,v) \) for all vertices \( v \) that are reachable from \( s \).

**Corollary 3** Let \( G = (V, E) \) be a weighted, directed graph with source vertex \( s \) and weight function \( w : E \to \mathbb{R} \). Then for each vertex \( v \in V \), there is a path from \( s \) to \( v \) if and only if \( \text{BELLMAN-FORD} \) terminates with \( d'[v] < \infty \) when it is run on \( G \).