Dynamic Programming

- Single Source Shortest Paths Problem
- Dynamic Programming Formulation
- Bellman-Ford Algorithm

Imdad ullah Khan

SSSP Problem

Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all vertices $v \in V$

This assumes no negative weight cycle in G

$\triangleright$ Actually, no negative cycle reachable from the source vertex s is sufficient

Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all $v \in V$ or output a negative cycle

SSSP Problem: Dynamic Programming Formulation

Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all $v \in V$ or output a negative cycle

- Focus on finding shortest paths (assume no negative weight cycle in G)
- Use the sequential nature of paths and use subpaths as subproblems
- How to define subproblems?

Bellman-Ford idea: Consider shortest paths with at most i edges

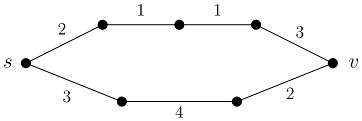
- Use hop lengths of paths, i to define sizes of problems
- i gives a natural ordering on the problems from larger to smaller
- To solve $\text{OPT}(v, i)$ we need to know only $\text{OPT}(u, j)$ for $j < i$
- Use shortest paths with at most i edges to find shortest paths with at most $i + 1$ edges
- Ordering subproblems become simpler, when we expand the number of subproblems

SSSP Problem: Dynamic Programming Formulation

Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all $v \in V$ or output a negative cycle

Focus on finding shortest paths (assume no negative weight cycle in G)

$\text{OPT}(v, i)$: minimum weight of a $s - v$ path that uses at most i edges



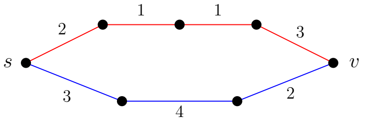
- If no $s - v$ path uses at most i edges, then $\text{OPT}(v, i) = \infty$
- If the best $s - v$ path uses $\leq i - 1$ edges then $\text{OPT}(v, i) = \text{OPT}(v, i - 1)$
- If the best $s - v$ path uses i edges and it's last edge is (u, v), then $\text{OPT}(v, i) = \text{OPT}(u, i - 1) + w(uv)$
- $d(v) := d(s, v) = \text{OPT}(v, n - 1)$

SSSP Problem: Dynamic Programming Formulation

Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all $v \in V$ or output a negative cycle

Focus on finding shortest paths (assume no negative weight cycle in G)

$\text{OPT}(v, i)$: minimum weight of a $s - v$ path that uses at most i edges



$d(v) = 7$
$\text{OPT}(v, 3) = 9$
$\text{OPT}(v, 2) = \infty$

- If no $s - v$ path uses at most i edges, then $\text{OPT}(v, i) = \infty$
- If the best $s - v$ path uses $\leq i - 1$ edges then $\text{OPT}(v, i) = \text{OPT}(v, i - 1)$
- If the best $s - v$ path uses i edges and it's last edge is (u, v), then $\text{OPT}(v, i) = \text{OPT}(u, i - 1) + w(uv)$
- $d(v) := d(s, v) = \text{OPT}(v, n - 1)$

SSSP Problem: Dynamic Programming Formulation

Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all $v \in V$ or output a negative cycle

$\text{OPT}(v, i)$: minimum weight of a $s - v$ path that uses at most i edges

To optimally reach v in at most i hops, do

1. optimally reach some neighbor u of v in at most $i - 1$ hops
2. hop from u to v

SSSP Problem: Dynamic Programming Formulation

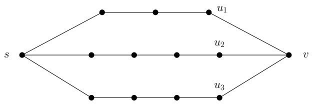
Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all $v \in V$ or output a negative cycle

$\text{OPT}(v, i)$: minimum weight of a $s - v$ path that uses at most i edges

If P is the best $s - v$ path using $\leq i$ edges and it's last edge is (u, v),
then $P \setminus \{(u, v)\}$ is the best $s - u$ path that uses at most $i - 1$ edges

Suppose P' is a path using $\leq i - 1$ edges and has $w(P') < w(P \setminus \{(u, v)\})$,
then $P' \cup \{(u, v)\}$ is a lighter path than P using $\leq i$ edge

$$\text{OPT}(v, i) = \min \begin{cases} \text{OPT}(v, i-1) \\ \min\limits_{u \in N(v)} \text{OPT}(u, i-1) + w(uv) \end{cases}$$



SSSP Problem: Dynamic Programming Formulation

Input: A weighted graph G and a source vertex $s \in V$
Output: Shortest paths from s to all $v \in V$ or output a negative cycle

$\text{OPT}(v, i)$: minimum weight of a $s - v$ path that uses at most i edges

$$\text{OPT}(v,i) = \min \begin{cases} \infty & \text{if } i = 0 \wedge s \neq v \\ 0 & \text{if } i = 0 \wedge s = v \\ \text{OPT}(v, i-1) & \\ \min\limits_{u \in N(v)} \text{OPT}(u, i-1) + w(uv) & \end{cases}$$

Maintain a table of solution for each v and $1 \leq i \leq n$ (memo)

A negative cycle is detected if $\text{OPT}(v, n)$ improves upon $\text{OPT}(v, n-1)$