Department of Industrial Engineering & Operations Research

IEOR 162 Linear Programming and Network Flows (Spring 2022)

Dijkstra's algorithm (1956)

Requirements

All arcs have non-negative cost. Cycles are allowed.

Key concepts

- At each iteration, the shortest path label $d(i)$ for node $i \in V$ represent the distance from the source s to node i with only permanent nodes as intermediate nodes.
- Once a node becomes permanent, its label $d(i)$ is the shortest path distance from s.

Input

- Graph $G = (V, A)$ with arc cost c_{ij}.
- Source node s.

Output

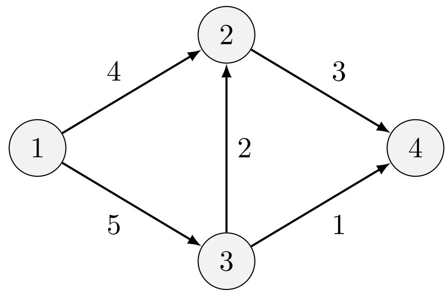
- $d(i)$: Distance from s to i for all $i \in V$.
- pred(i): Predecessor of node i for all $\in V$.

Steps

1. The set of temporary nodes is initialized as the set of all nodes; $T = \{1, \ldots, n\}$.
2. The set of permanent nodes is initialized as the empty set; $P = \emptyset$.
3. Set $d(s) = 0$ and $d(i) = \infty$ for $i \in V - \{s\}$.
4. Select the temporary node with the smallest label:
$$i^* = \arg\min_{i \in T}\{d(i)\}.$$
5. Mark node i^* as permanent:
$$P \leftarrow P \cup \{i^*\},\ T \leftarrow T - \{i^*\}.$$
6. Update the distance (and predecessor) for each temporary out-neighbor of i^* :
$$d(j) \leftarrow \min\{d(j), d(i^*) + c_{i^*j}\} \quad \text{for each } (i^*, j) \in A, j \in T.$$
7. Repeat steps 4 to 6 until all nodes are permanent.

Example

We want to find the shortest paths from 1 to all other nodes.



Initialization: Iteration 0

$P = \emptyset$
$T = \{1, 2, 3, 4\}$

Node	1	2	3	4
Iter 0	0(-)	∞ (-)	∞ (-)	∞ (-)

Iteration 1

Node 1 has the smallest label and becomes permanent (denoted by $*$).
$P = \{1\}$
$T = \{2, 3, 4\}$

Neighbor update:

Node 2: $d(2) \leftarrow \min\{\infty, 0 + 4\} = 4$, $\mathrm{pred}(2) = 1$.

Node 3: $d(3) \leftarrow \min\{\infty, 0 + 5\} = 5$, $\mathrm{pred}(3) = 1$.

Node	1	2	3	4
Iter 0	$0(-)$	$\infty(-)$	$\infty(-)$	$\infty(-)$
Iter 1	$0^*(-)$	$4(1)$	$5(1)$	$\infty(-)$

Iteration 2

Node 2 has the smallest label and becomes permanent.
$P = \{1, 2\}$
$T = \{3, 4\}$

Neighbor update:

Node 4: $d(2) \leftarrow \min\{\infty, 4 + 3\} = 7$, $\text{pred}(4) = 2$.

Node	1	2	3	4
Iter 0	$0(-)$	$\infty(-)$	$\infty(-)$	$\infty(-)$
Iter 1	$0^*(-)$	$4(1)$	$5(1)$	$\infty(-)$
Iter 2		$4^*(1)$	$5(1)$	$7(2)$

Iteration 3

Node 3 has the smallest label and becomes permanent.
$P = \{1, 2, 3\}$
$T = \{4\}$

Neighbor update:

Node 4: $d(4) \leftarrow \min\{7, 5 + 1\} = 6$, $\text{pred}(4) = 3$.

The distance for node 2 isn't updated since it is a permanent node.

Node	1	2	3	4
Iter 0	$0(-)$	$\infty(-)$	$\infty(-)$	$\infty(-)$
Iter 1	$0^*(-)$	$4(1)$	$5(1)$	$\infty(-)$
Iter 2		$4^*(1)$	$5(1)$	$7(2)$
Iter 3			$5^*(1)$	$6(3)$

Iteration 4

Node 4 has the smallest label and becomes permanent.
$P = \{1, 2, 3, 4\}$
$T = \emptyset$

Neighbor update:
There are no out-neighbors

Node	1	2	3	4
Iter 0	$0(-)$	$\infty(-)$	$\infty(-)$	$\infty(-)$
Iter 1	$0^*(-)$	$4(1)$	$5(1)$	$\infty(-)$
Iter 2		$4^*(1)$	$5(1)$	$7(2)$
Iter 3			$5^*(1)$	$6(3)$
Iter 4				$6^*(3)$