Algorithms Theory

11 – Shortest Paths

Dr. Alexander Souza

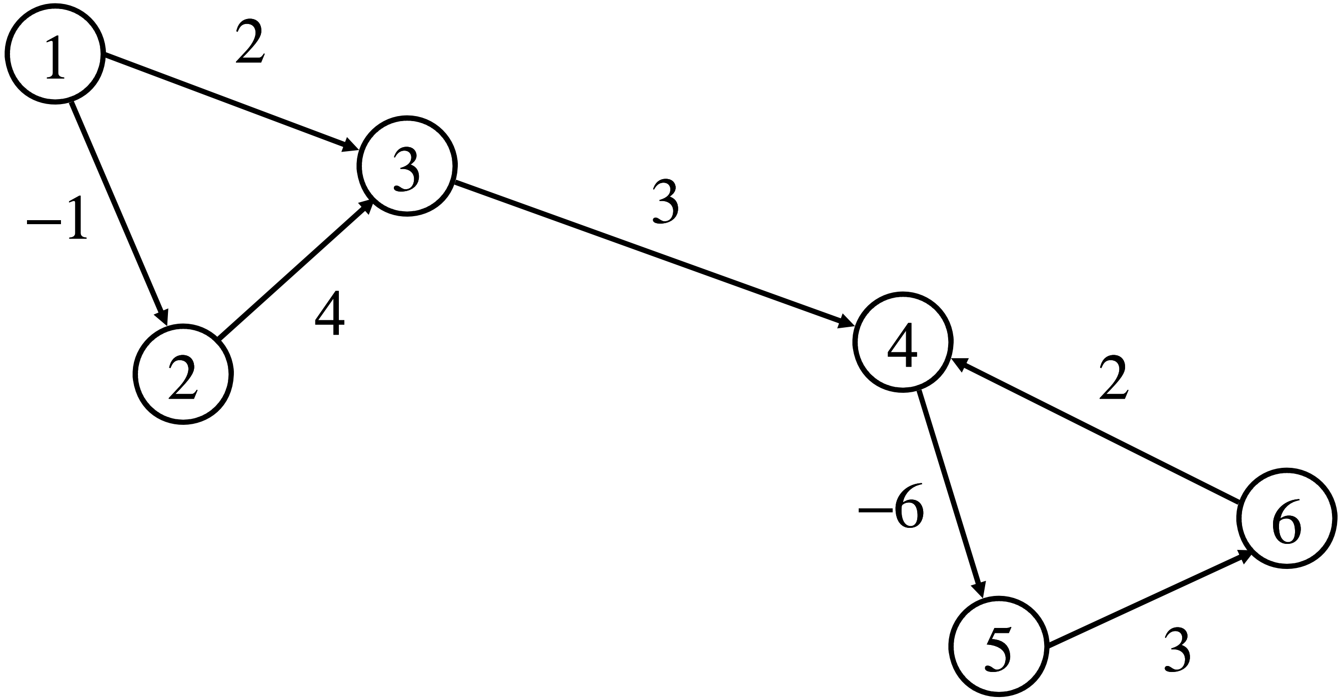
Wed, 25.1, No lecture
Wed, 1.3, 90 lecture

Winter term 11/12

1. Shortest-paths problem

Directed graph $G = (V, E)$

Cost function $c : E \rightarrow R$



Distance between two vertices

Cost of a path $P = v_0, v_1, \ldots, v_l$ from u to v : $v_i\, v_{i+1} \in E$

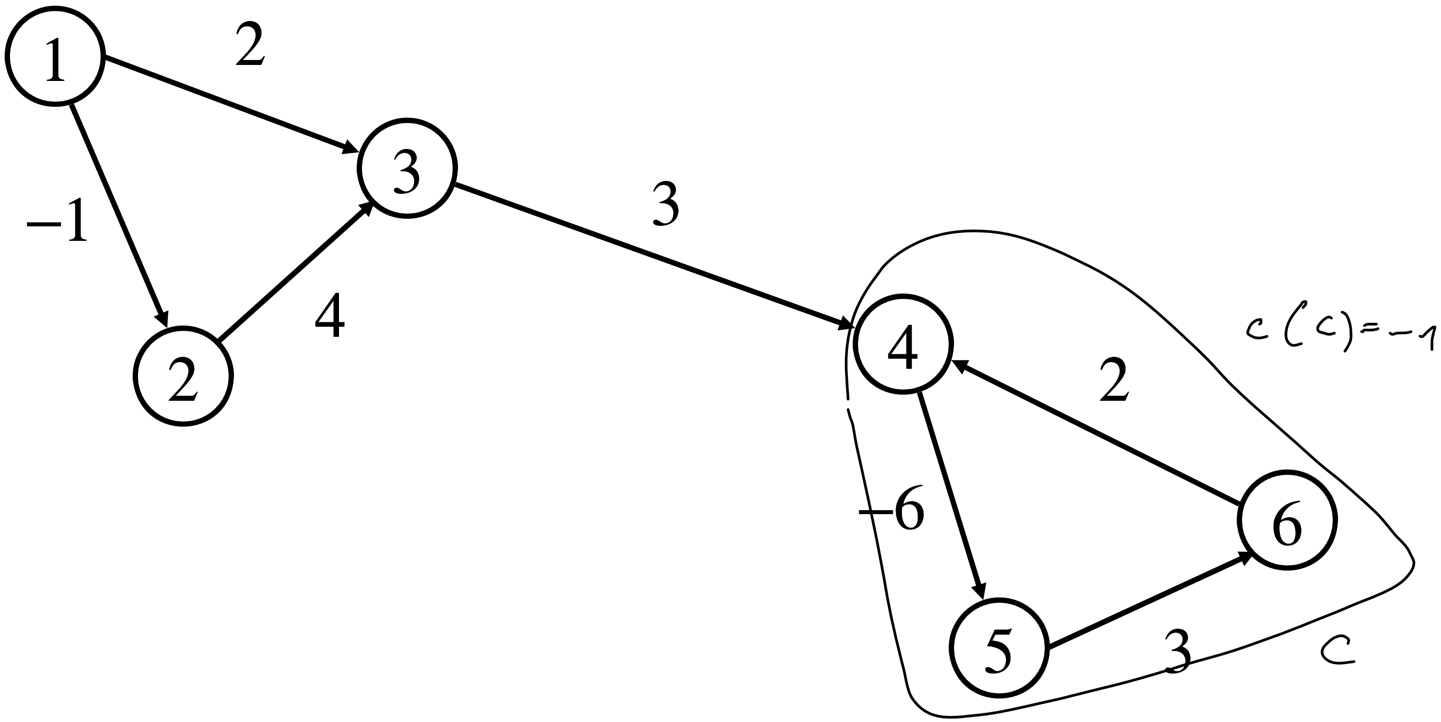
$u = v_0$, $v = v_l$

$$c(P) = \sum_{i=0}^{l-1} c(v_i, v_{i+1})$$

Distance between u and v (not always defined):

$dist(u,v) = \inf \{ c(P) \mid P \text{ is a path from } u \text{ to } v \}$

Example



$dist(1,2) =$

$dist(1,3) =$

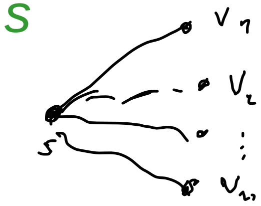
$dist(3,1) = +\infty$

$dist(3,4) = -\infty$

2. Single-source shortest paths problem

Input: network $G = (V, E, c)$, $c : E \to R$, vertex s

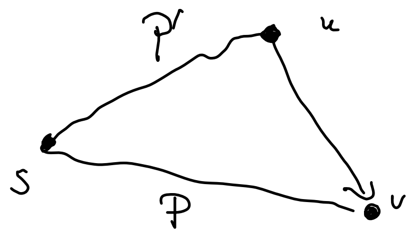
Output: $dist(s,v)$ for all $v \in V$

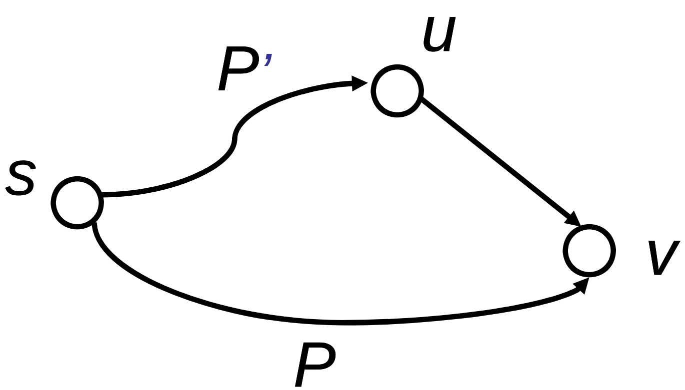


Observation: The function *dist* satisfies the triangle inequality.

For any edge $(u,v) \in E$:

$$dist(s,v) \leq dist(s,u) + c(u,v)$$





P = shortest path from s to v

P' = shortest path from s to u

Greedy approach to an algorithm

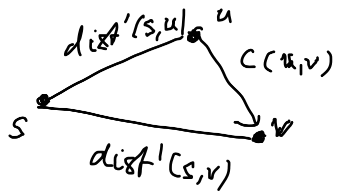
1. Overestimate the function *dist*

$$dist'(s,v) = \begin{cases} 0 & \text{if } \; v = s \\ \infty & \text{if } \; v \neq s \end{cases}$$

2. While there exists an edge $e = (u,v)$ with

$$dist'(s,v) > dist'(s,u) + c(u,v)$$

set $dist'(s,v) \leftarrow dist'(s,u) + c(u,v)$



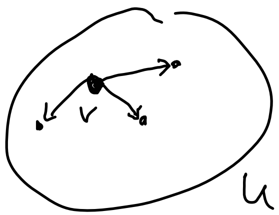
Generic algorithm

1. DIST[s] $\leftarrow 0$;
2. **for all** $v \in V \setminus \{s\}$ **do** DIST[v] $\leftarrow \infty$ **endfor;**
3. **while** $\exists\, e = (u,v) \in E$ with DIST[v] > DIST[u] + $c(u,v)$ **do**
4. Choose such an edge $e = (u,v)$;
5. DIST[v] $\leftarrow$ DIST[u] + $c(u,v)$; restore triangle ineq.
6. **endwhile;**

Questions:

1. How can we efficiently check in line 3 if the triangle inequality is violated?
2. Which edge shall we choose in line 4?

Solution



Maintain a set U of all those vertices that might have an outgoing edge violating the triangle inequality.

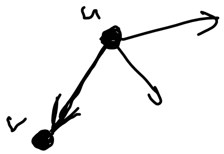
- Initialize $U = \{s\}$
- Add vertex v to U whenever DIST[v] decreases.

$$\text{DIST}[w] \leq \text{DIST}[v] + c(v,w)$$

1. Check if the triangle inequality is violated: $U \neq \emptyset$?
2. Choose a vertex from U and restore the triangle inequality for all outgoing edges (edge relaxation).

Refined algorithm

1. DIST[s] $\leftarrow 0$;
2. **for all** $v \in V \setminus \{s\}$ **do** DIST[v] $\leftarrow \infty$ **endfor;**
3. $U \leftarrow \{s\}$;
4. **while** $U \neq \emptyset$ **do**
5. Choose a vertex $u \in U$ and delete it from U;
6. **for all** $e = (u,v) \in E$ **do**
7. **if** DIST[v] > DIST[u] + $c(u,v)$ **then**
8. DIST[v] $\leftarrow$ DIST[u] + $c(u,v)$; $\leftarrow$ restore $\triangle$-ineq.
9. $U \leftarrow U \cup \{v\}$;
10. **endif;**
11. **endfor;**
12. **endwhile;**



Invariant for the DIST values

Lemma 1: For each vertex $v \in V$ we have DIST[v] $\geq$ $dist(s,v)$.

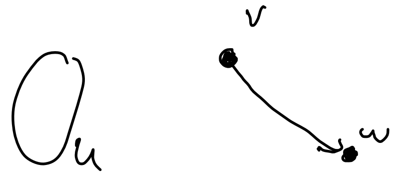
Proof: (by contradiction)

Let v be the first vertex for which the relaxation of an edge (u,v) yields DIST[v] $<$ $dist(s,v)$.

Then:

$$\text{DIST}[u] + c(u,v) = \text{DIST}[v] < dist(s,v) \leq dist(s,u) + c(u,v)$$

Important properties



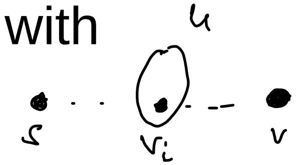
Lemma 2:

a) If $v \notin U$, then for all $(v,w) \in E$: DIST[w] $\leq$ DIST[v] + $c(v,w)$

U serves its purpose.

b) Let $s = v_0, v_1, ..., v_l = v$ be a shortest path from s to v. l finite

If DIST[v] > $dist(s,v)$, then there exists v_i, $0 \leq i \leq l$ -1, with $v_i \in U$ and DIST[v_i] = $dist(s,v_i)$.



$\rightsquigarrow$ $U \neq \emptyset$ $\Rightarrow$ Alg. has not yet terminated

c) If G has no negative-cost cycles and DIST[v] > $dist(s,v)$ for any $v \in V$, then there exists a $u \in U$ with DIST[u] = $dist(s,u)$.

Shortest path ~~cost~~ length finite $\overset{\rightsquigarrow}{b)}$ set $u = v_i$.

d) If in line 5 we always choose $u \in U$ with DIST[u] = $dist(s,u)$, then the while-loop is executed only once per vertex.

Such a vertex u leaves U but never re-enters.

Lemma 1.

Proof of Lemma 2

a) Induction on the number *i* of executions of while-loop

i = 0:

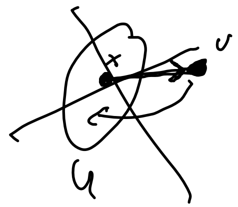
Vertices $v \neq s$ are not in U

$$\mathrm{DIST}[w] \leq \underbrace{\mathrm{DIST}[v]}_{\infty} + c(v, w)$$

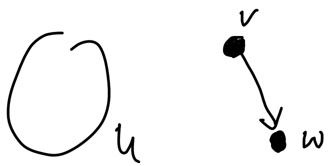
Proof of Lemma 2

i > 0: Assume that statement holds before i-th execution of the while loop. We show that it holds afterwards.

$v \notin U$ after i-th execution of while loop



~~Case 1~~ Case 1: $v \notin U$ before i-th exe.



$$\text{DIST}[w] \leq \text{DIST}[v] + c(v,w)$$

does not change

can only decrease

$\Rightarrow$ $\triangle$ for (v,w) holds after i-th exe.

Case 2: $v \in U$ before i-th exe.

For each (v,w) which $\text{DIST}[w] > \text{DIST}[v] + c(v,w)$

we set $\text{DIST}[w] = \text{DIST}[v] + c(v,w)$.

And we remove v from U.

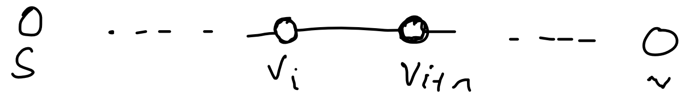
Proof of Lemma 2

b) $P: s = v_0, v_1, \dots, v_\ell = v$

Let i be max. with $DIST[v_i] = dist(s, v_i)$.

$i = 0$ satisfies this condition because $\ell < \infty$, i.e. s is not on a neg. cost cycle. Thus i exists.

Assume that $v_i \notin U$.



$$DIST[v_{i+1}] \underset{(a)}{\leq} DIST[v_i] + c(v_i, v_{i+1})$$

$$= dist(s, v_i) + c(v_i, v_{i+1})$$

$$= dist(s, v_{i+1})$$

$\Longrightarrow DIST[v_{i+1}] = dist(s, v_{i+1})$ ↯

Contradicts maximality of i.