Nov 29

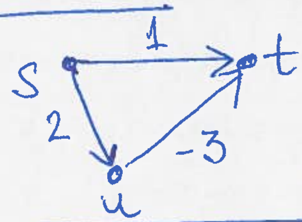
Shortest Path Problem

<u>Input</u>: (•) Directed graph $G=(V,E)$; $\forall e \in E$, cost c_e (can be <0) but NO negative cycle
(•) $t \in V$

<u>Output</u>: $\forall s \in V$, a shortest s-t path ($c(P) = \sum_{e \in P} c_e$, min $c(P)$)

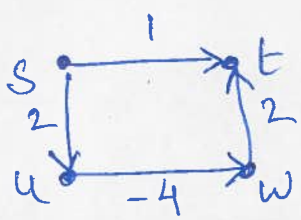
NOTE: Unlike Dijkstra, we have one destination.

<u>ATTEMPT 1</u>: Run Dijkstra from each $s \in V$ (Fails miserably)

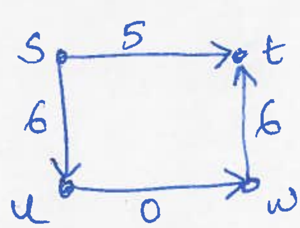


→ start Dijkstra at S
$d'(t) = 1$, $d(u) = 2$ ⇒ Dijkstra sets $d(t) = 1$
{s,t} is a shortest path.
WRONG: shortest s-t path s, u, t.

<u>ATTEMPT 2</u>: Try a "smarter" reduction to Dijkstra
→ Add a large enough number to all edges so that all the new edge costs $c'_e \geq 0$. Run Dijkstra on new instance



Add 4 to all edges →



shortest s-t path: s, u, w, t

shortest s-t path: s, t

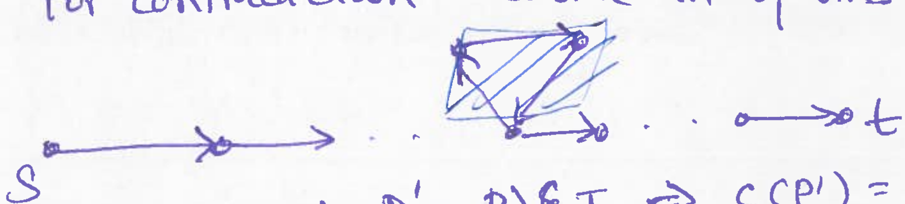
→ No known greedy algo

<u>Bellman-Ford</u> (one of the first dynamic programming algo)

<u>PROPOSITION</u>: If G has no negative cycle ⇒ $\forall$ s,t $\exists$ a simple shortest s-t path.

<u>Pf (idea)</u>: For contradiction assume all s-t paths have a cycle in them

P :



⇒ Just remove the cycle.

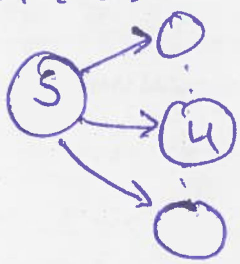
$\exists$ a cycle T, let $P' = P \setminus \{T\}$ ⇒ $c(P') = c(P) - c(T)$
$\leq c(P)$ {as no negative cycle $\exists$}

→ Repeat if necessary to get a simple path with cost $\leq c(P)$ ∎

ASSUME: Only interested in cost of shortest paths

ATTEMPT 3: Use "weighted interval scheduling like" dynamic prog.

$OPT(s)$ = cost of shortest s-t path. ($\forall s$)



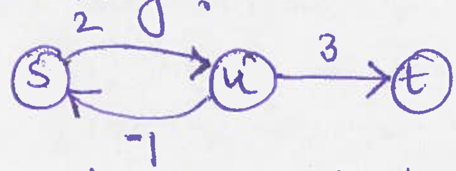
If a shortest s-t path takes edge (s,u)

$$OPT(s) = C_{(s,u)} + OPT(u)$$

In general: $OPT(s) = \min_{u:(s,u)} C_{(s,u)} + OPT(u)$

① # sub-probems: n ✓ ② recurrence formula: ↓ ✓

③ Ordering?



$OPT(s) = 2 + OPT(u)$

$OPT(u) = \min\{3 + OPT(t), -1 + OPT(s)$

PROBLEM! $OPT(s)$ depends on $OPT(u)$ } ⇒ no hope of a total ordering
$OPT(u)$ ——— (s)

ATTEMPT 4! Like subset sum let us use ALL parameters

E.g. $OPT(s, G)$: cost of shortest s-t path in G.

$$OPT(s, G) = C(s,u) + OPT(u, G\setminus\{(s,u)\})$$

or more generally: $OPT(s,G) = \min_{u:(s,u) \in E} \{C_{(s,u)} + \cdot\}$

→ solves the ordering issue.

PROBLEM! If you "unroll" the recurrence we need to compute $OPT(s, (V, E')) \ \forall E' \subseteq E$

⇒ exponentially many sub-problems!

ATTEMPT 5: Idea: Think of a measure of how "close" we are to t.

$OPT(s,i)$ = cost of shortest s-t path with $\le i$ edges.