Lecture18_ShortestPaths2

Tuesday, October 20, 2020 4:21 PM

Unweighted graph

Suppose that $w(u, v) = 1$ for all $(u, v) \in E$. Can Dijkstra's algorithm be improved?

The PQ we have
improved path, n't
have dramatic c
➢ Use a sim

Correctness of BFS

while $Q \neq \emptyset$
  do $u \leftarrow$ DEQUEUE(Q)
    for each $v \in Adj[u]$
      do if $d[v] = \infty$
        then $d[v] \leftarrow d[u] + 1$
          ENQUEUE(Q, v)

Key idea:
The FIFO Q in breadth-first search mimics the priority queue Q in Dijkstra.
- **Invariant:** v comes after u in Q implies that $d[v] = d[u]$ or $d[v] = d[u] + 1$.

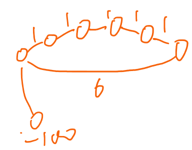
Running time: O(V + E)

- W
- Li

Determi

Question: remove negative edge weight by adding x t
smallest edge weight in the graph

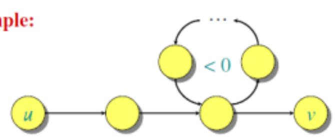
➢ Do we still have the same problem?? Do they h
➢ Design a counter example showing that this ide
No! the problem changes!! The shortest path changes!!
The addition may affect the path multiple times
- This modification based on for each path, we add number of edges times x to the shortest path.



Negative-weight cycles

If a graph G = (V, E) contains a negative weight cycle, then some shortest paths may not exist.

Example:



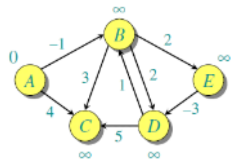
Bellman-Ford algorithm: Finds all shortest-path lengths from a source s ∈ V to all v ∈ V or determines that a negative-weight cycle exists.
- Figure out if there exist negative cycles

$d[s] \leftarrow 0$
for each $v \in V - \{s\}$ } initialization
  do $d[v] \leftarrow \infty$

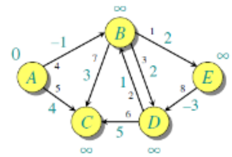
for $i \leftarrow 1$ **to** $|V| - 1$
  do for each edge $(u, v) \in E$
    do if $d[v] > d[u] + w(u, v)$ } *relaxation step*
      then $d[v] \leftarrow d[u] + w(u, v)$
for each edge $(u, v) \in E$
  do if $d[v] > d[u] + w(u, v)$
    then report that a negative-weight cycle exists

At the end, $d[v] = \delta(s, v)$, if no negative-weight cycles.
Time = $O(VE)$.

// do relaxation when found better path
How many times we do relaxation? In which order we do?
Does not matter (matters for Dijkstra)
N - 1 pass maximum. (at most n - 1 edges for n vertices)
After n -1 relaxations, if it still can do relaxations, then there is negative cycle.*



Initialization.



Order of edge relaxation.

- One node can be update more than one time in single i
- If one whole pass doesn't change any ting, done

Correctness:

Theorem. If G = (V, E) contains no negative weight cycles, then after the Bellman-Ford algorithm executes, d[v] = δ(s, v) for all v ∈ V.

Corollary. If a value d[v] fails to converge after |V| – 1 passes, there exists a negative-weight cycle in G reachable from s.