Negative Weights

Djikstra: doesn’t work with negative weights

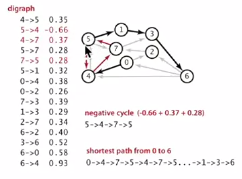


Reweighting: add a constant to every edge-> this doesn’t work



Need a different algorithm (topological sort doesn’t work, as a cycle may be present)

Negative cycle: a directed cycle whose sum of edge weights is negative



Proposition: A SPT exists iff no negative cycles

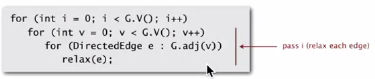
Bellman-Ford algorithm

Initialize distTo[s] = 0 and distTo[v] = infinity for all other vertices

Repeat V times:

* Relax each edge

Implementation



Proposition: dynamic programming algorithm computes SPT in any edg-weighted digraph with no negative cycles in time proportional to E x V

Proof idea: after pass *I*, found shortest path containing at most I edges

Observation: If distTo[v] does not change during pass I, no need to relax any edge pointing from v in pass i+1

FIFO implementation: maintain queue of vertices whose distTo[] changed  
 *-> be careful to keep at most one copy of each vertex on queue*

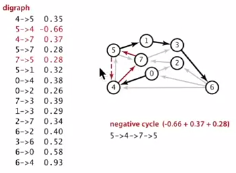
Overall effect:

* The running time is still proportional to E x V in the worst case
* But much faster than that in practice

Finding a negative cycle

Add two methods to the API for SP

Boolean hasNegativeCycle() : is there a negative cycle?  
Iterable<DirectedEdge> negativeCycle() : negative cycle reachable from s



Observation: if there is a negative cycle, Bellman-Ford gets stuck in loop, updating distTo[] and edgeTo[] entries of vertices in the cycle

Proposition: if any vertex v is updated in phase v, there exists a negative cycle (and can trace back edgeTo[v] entries to find it)

In practice: check for negative cycles more frequently

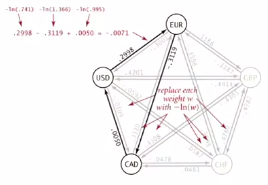
Negative cycle application: arbitrage detection:

Problem: given table of exchange rates, is there an arbitrage opportunity?



Ex: $1,000 -> 741 Euros -> $CA 1,012.206 -> $1,007.14497

Currency exchange graph:

* Vertex = currency
* Edge = transaction, with weight equal to exchange rate
* Find a directed cycle whose product of edge weights is > 1

Model as a negative cycle detection problem by taking logs:

* Let weight of edge v -> w be – ln (exchange rate from currency v to w)
* Multiplication turns to addition; > 1 to < 0
* Find a directed cycle whose sum of edge weights is < 0 (negative cycle)

Fastest algorithm is extremely valuable!