Shortest Paths 2

October 10, 2008

Some contents of this presentation are adapted from year 2005 course notes for 620-362 Applied Operations Research, Department of Mathematics and Statistics, The University of Melbourne (compiled by Prof Natashia Boland and Dr Renata Sotirov)

Outline

So Far

We aim to find the least cost path between two specified nodes s and t in a network.

We can do this if all costs are positive (Dijkstra's Algorithm)

OR

If the network is acyclic (Reaching Algorithm)

What if the network has cycles and negative cost arcs?

So Far

We aim to find the least cost path between two specified nodes s and t in a network.

We can do this if all costs are positive (Dijkstra's Algorithm)

OR

If the network is acyclic (Reaching Algorithm)

What if the network has cycles and negative cost arcs?

So Far

We aim to find the least cost path between two specified nodes *s* and *t* in a network.

We can do this if all costs are positive (Dijkstra's Algorithm)

OR

If the network is acyclic (Reaching Algorithm)

What if the network has cycles and negative cost arcs?

So Far

We aim to find the least cost path between two specified nodes s and t in a network.

We can do this if all costs are positive (Dijkstra's Algorithm)
OR
If the network is acyclic (Reaching Algorithm)

What if the network has cycles and negative cost arcs?

From Here

A general method for any graph without negative cost cycles.

Two ways of detecting negative cost cycles.

Modelling in a shortest path.

Outline

Notation

$d(i)$ = length of the shortest path found so far to node $i \in N$ from node s
$k(i)$ = immediate predecessor of i on the current shortest path from s to i

Initialisation
$d(s) = 0 \qquad d(i) = \infty \quad \forall i \in N \setminus \{s\}$
$k(i)$ is not defined

A Basic Label-Correcting Algorithm

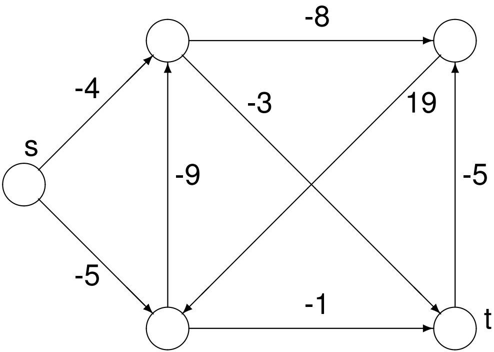
while for some arc $(i, j) \in A \quad d(j) > d(i) + c_{i,j}$ **do**

$d(j) := d(i) + c_{i,j}$

$k(j) := i$

end while

Using the Basic Algorithm



Using a Queue

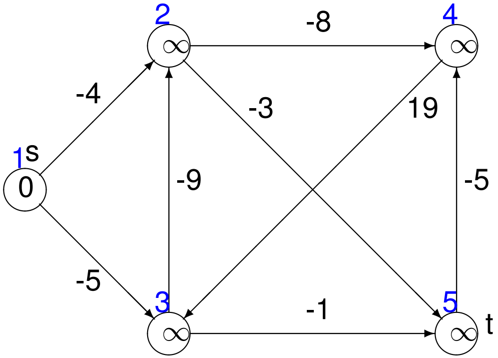
Maintain an ordered list Q of nodes and investigate the arcs out of the first node in Q.

$Q = (s)$

Using a Queue

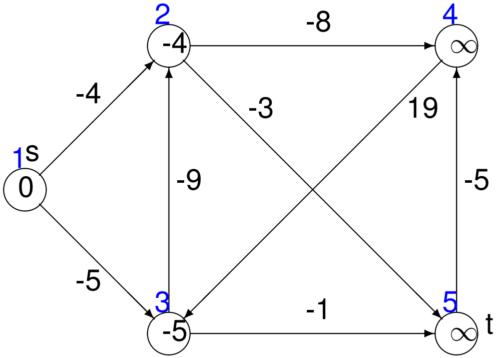
while $Q \neq \emptyset$ **do**
  Let i be the first node in Q
  Remove i from Q
  for $j \in \delta(i)$ **do**
    if $d(j) > d(i) + c_{i,j}$ **then**
      $d(j) := d(i) + c_{i,j}$
      $k(j) := i$
      if j has never been in Q **then**
        add j to the end of Q
      else
        add j to the start of Q
      end if
    end if
  end for
end while

Using a Queue



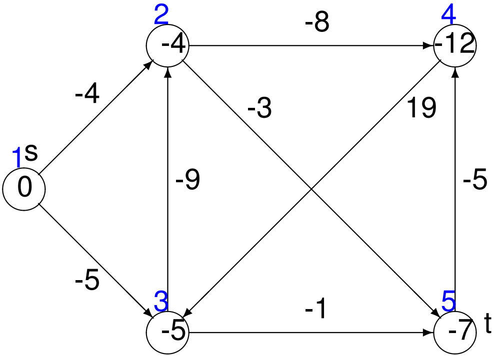
$Q = [1]$

Using a Queue



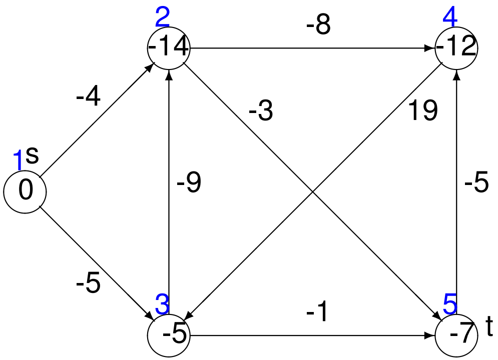
$Q = [2, 3]$

Using a Queue



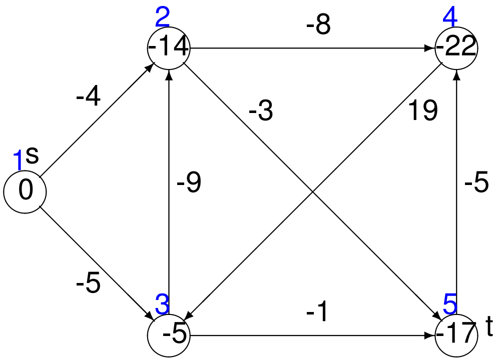
$Q = [3, 4, 5]$

Using a Queue



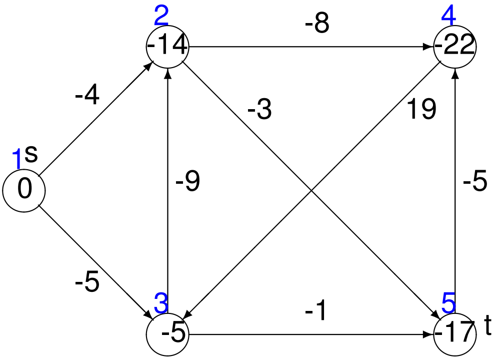
$Q = [2, 4, 5]$

Using a Queue



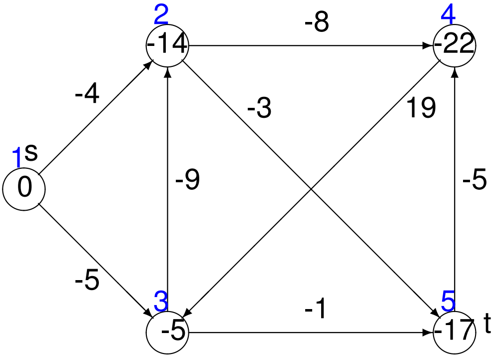
$Q = [4, 5]$

Using a Queue



$Q = [5]$

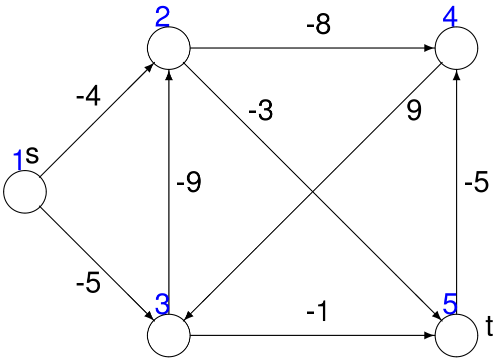
Using a Queue



$Q = [\emptyset]$

Outline

Using a Queue



Detecting Negative Cycles via Total Cost

We can find a lower bound on the cost of any path to any node in the network as follows:

Let $A^{-} = \{(i,j) \in A | c_{i,j} < 0\}$
Let $M = \sum_{(i,j) \in A^{-}} c_{i,j}$
Now M is our lower bound.
If any node has a cost lower than M then we can stop the algorithm as we have detected a negative cost cycle.
In our example, $M = -35$

Detecting Negative Cycles via Total Cost

We can find a lower bound on the cost of any path to any node in the network as follows:

Let $A^- = \{(i,j) \in A | c_{i,j} < 0\}$
Let $M = \sum_{(i,j)\in A^-} c_{i,j}$
Now M is our lower bound.
If any node has a cost lower than M then we can stop the algorithm as we have detected a negative cost cycle.
In our example, $M = -35$

Detecting Negative Cycles via Total Cost

We can find a lower bound on the cost of any path to any node in the network as follows:

Let $A^{-} = \{(i,j) \in A | c_{i,j} < 0\}$
Let $M = \sum_{(i,j) \in A^{-}} c_{i,j}$
Now M is our lower bound.
If any node has a cost lower than M then we can stop the algorithm as we have detected a negative cost cycle.
In our example, $M = -35$

Detecting Negative Cycles via the Shortest Path Tree

We can keep track of the shortest path tree implied by the labelling procedure.
If at any stage, the 'tree' has a cycle, then this is a negative cost cycle.

Outline

Equipment Replacement Strategy

It costs $19,000 to buy a new car.

Each year a car becomes more expensive to run.

Age Of Car	Annual Costs	Trade-In Value
0	$1,000	$12,000
1	$2,000	$10,000
2	$5,000	$6,000
3	$9,000	$5,000
4	$12,000	$3,000

We aim to determine the best strategy for the next 5 years.

Equipment Replacement Strategy

New car prices

Start Of Year	Price
1	$19,000
2	$18,500
3	$18,000
4	$18,000
5	$17,000