NETWORK OPTIMIZATION

Megan Bryant
mrbryant@email.wm.edu

Department of Mathematics, The College of William and Mary, P.O. Box 8795, Williamsburg, VA 23187

Executive Summary

Homework 5.
Due date: March 3rd, at the start of class
Collaborators:

1

Implement the FIFO modified label correcting algorithm. Your algorithm should take a directed graph with arc costs c as input, plus a source node s and a sink node t. The algorithm should detect whether a negative cycle exists, and if not, it should output the shortest path from node s to node t. If a negative cycle exists, your algorithm should output a negative cycle.

We know that the FIFO algorithm is a specific implementation of the modified label-correcting algorithm. It maintains the set of LIST as a queue and hence examines nodes in the LIST in a first-in, first-out order (queue order). It has $O(nm)$ time and can identify the presence of negative cycles by recording the number of times that the algorithm examines each node. If the network contains no negative cycles, then the algorithm will examine each node at most $n-1$ times. If the algorithm examines a node more than $n-1$ times, then the network must have a negative cycle.

FIFO Modified Label-Correcting

begin
  $d(s) := 0$ and $\text{pred}(s) := 0$
  $d(j) := \infty$ for each node $j \in N - \{s\}$
  QUEUE := $\{s\}$
  while QUEUE $\neq \emptyset$ **do**
  begin remove an element i from LIST;
    for each arc $(i, j) \in A(i)$ **do**
    if $d(j) > d(i) + c_{ij}$;
    begin
      $d(j) := d(i) + c_{ij}$;
      $\text{pred}(j) := i$;
      if $j \notin$ LIST **then** add node j to LIST;
    end;
  end;
end;

Note that this algorithm is nearly identical to modified label-correcting algorithm. The difference between the two is that in the FIFO implementation, the list is maintained as a queue. This ensures that the nodes in the list are examined in first-in, first-out order (FIFO). The algorithm is therefore much more efficient in practice.

The following implementation of the FIFO algorithm was developed in MatLab.

FIFO Algorithm (MatLab Code):

```
function [path, totalCost] = fifo(A, s, t)
n = size(A,1);

d(1:n) = inf;
d(s) = 0;
pred(1:n) = 0;
LIST = [s];
count(1:n) = 0;

while not(isempty(LIST))
i = LIST(1);
if (count(i) > (2))
display(['Negative cycle detected with node ' num2str(i)]);
break;
end;
count(i) = count(i) + 1;
for j = (1:n)
if (d(j) > d(i) + A(i,j))
d(j) = d(i) + A(i,j);
pred(j) = i;
if (ismember(j,LIST)==0)
LIST(end+1) = j;
end;
end;
end;
LIST(1) = [];
end;


path = [];
if pred(t) ~= 0
i = t;
path = [t];
while i ~= s
p = pred(i);
path = [p path];
i = p;
end;
end;

totalCost = d(t);
display(d(t));
return;
```

This implementation was tested against the following generic cost matrix with a negative cost cycle starting at node 1.

$$\begin{pmatrix} 0 & -2 & Inf \\ Inf & 0 & 1 \\ -1 & Inf & 0 \end{pmatrix}$$

When the implementation was tasked with finding the shortest path between nodes 1 and 3, the result was the following output:

```
EDU>> fifo(test,1,3)
Negative cycle detected with node 1

ans =

-5


ans =

1 2 3
```

We see that the FIFO implementation has successfully detected the presence of a negative cycle originating at node 3. It is therefore able to terminate appropriately.

2

In currency exchange, an arbitrage opportunity exists, if it is possible to make a sequence of exchanges so that the amount of money you end with is larger than the amount of money with which you started. For example, in Figure 1, you could exchange DM for Y, then Y for F, and F for DM, and end up with $56 \times \frac{3}{50} \times \frac{3}{10} = 1.008$ *times the amount you started with.*

Formulate the problem of deciding whether arbitrage is possible as the problem of detecting a negative cost cycle. Use the algorithm you implemented in 1, to make sure your method correctly identifies there is an arbitrage opportunity in Figure 1. Hint: $ab > 1$ *is equivalent to* $\log(ab) = \log a + \log b > 0$.

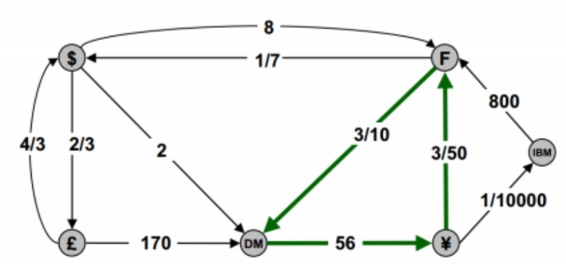

Figure 1: An arbitrage opportunity is indicated by the green cycle.

We would like to formulate the problem of detecting arbitrage opportunities in terms of the detection of a negative cycle.

We know that an arbitrage opportunity arises when we have a positive return on investment. That is to say, given for any two ROIs a and b, an arbitrage opportunity exists in our original formulation if $ab > 1$. The FIFO implementation however, calculates shortest paths using addition. Therefore, if we are to formulate this arbitrage detection problem in terms of the FIFO negative cycle detection, we must translate this multiplication relationship in terms of addition. This is best accomplished with the use of logarithms.

$$\begin{aligned} ab &> 1 \\ \log(ab) &> \log(1) \\ \log(a) + \log(b) &> 0 \\ -\log(a) - \log(b) &< 0 \end{aligned}$$

The final transformation (multiplication by -1) is necessary if we are to successfully formulate the problem as a FIFO negative cycle detection. Therefore, in order to create an input graph for FIFO from the arbitrage detection problem, we should generate a cost matrix.

For every arc (i, j) with c_{ij} in the original formulation, the entry in the cost matrix is $-\log(c_{ij})$. In addition, we must add arcs with a cost of ∞ such that the graph is connected. The high cost ensures that they will never be selected for the shortest path, but allows the FIFO algorithm to proceed correctly. The cost matrix for the example is given below.

$$\begin{pmatrix} Inf & -\log(8) & -\log(\frac{2}{3}) & -\log(2) & Inf & Inf \\ -\log(\frac{1}{7}) & Inf & Inf & -\log(\frac{3}{10}) & Inf & Inf \\ -\log(\frac{4}{3}) & Inf & Inf & -\log(170) & Inf & Inf \\ Inf & Inf & Inf & Inf & -\log(56) & Inf \\ Inf & -\log(\frac{3}{50}) & Inf & Inf & Inf & -\log(\frac{1}{10000}) \\ Inf & -\log(800) & Inf & Inf & Inf & Inf \end{pmatrix}$$

Given this cost matrix, we can determine if there exists an arbitrage opportunity by using out FIFO algorithm to detect negative cost cycles. This is done by searching for a shortest path from a node i back to that node i. Since profitable exchanges have negative arc costs, the algorithm will attempt to find a negative distance shortest path.

We tested this formulation on the above input. The implementation began at node 4 and attempted to detect a negative cost cycle. It was able to do so successfully.

```
Algorithm Output:
EDU>> fifo(arbitrage,4,2)
Negative cycle detected with node 4

ans =

-5.2083


ans =

4 5 2
```

We know that the detection of a negative cost cycle implies the opportunity for currency arbitrage since that implies that there exists a path in the graph G such that the return on investment is greater than one. Therefore, we know from our formulation that if a negative cost cycle is detected, then there exists an opportunity for currency arbitrage. We saw this demonstrated with our example as a negative cost cycle was appropriately detected when FIFO found the shortest path from 4 to 2.

This detection of a negative cost cycle is correct in that, by our original formulation, the detection of a negative cost cycle occurs when for any two arc costs, a and b, $-\log(a) - \log(b) < 0$. This is exactly the case when an opportunity for currency arbitrage exists in the original problem as it is equivalent to $ab > 0$. Therefore, we know that we can correctly formulate the currency arbitrage problem as the negative cycle detection problem and can correctly detect when opportunities for arbitrage exist.