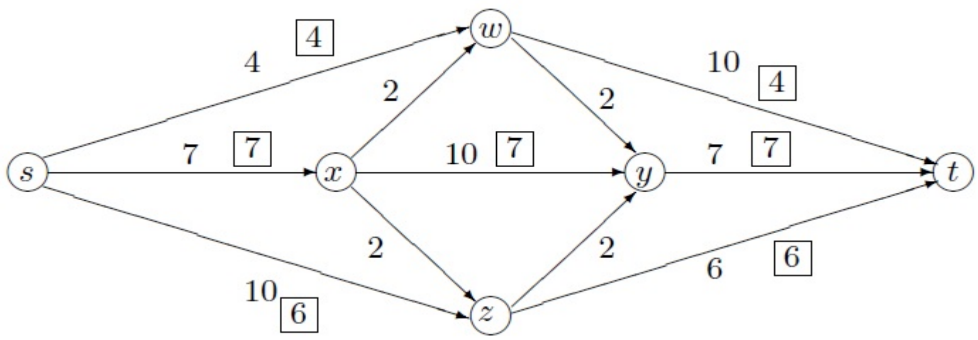
CS 6385.0W2 - Algorithmic Aspects of Telecommunication Networks - Su19

Assignment 2

Submitted By: ***Yash Pradhan (ypp170130)***

Problem Statement:



(a) What is the value of this flow?
(b) Is this a maximum s-t flow in this graph? If not, find a maximum s-t flow.
(c) Find a minimum s-t cut. (Specify which vertices belong to the sets of the cut.)

1. *Solve using formulate it as an LP problem.*
2. *Use Ford-Fulkerson Algorithm.*
3. *Compare the two results*

Solving it by formulating it as an LP Problem.

YPP170130

LP Formulation of Max Flow Problem:

- Objective:

maximize F $= \sum_{j=1}^{n} x_{sj} - \sum_{K=1}^{n} x_{KS}$

(outflow) (Inflow)

x_{ij} : denotes flow on link $i \to j$.

$(i) \xrightarrow[c_{ij}]{x_{ij}} (j)$

c_{ij} : Capacity of link $i \to j$

- Constraints:

$\forall i,j \,;\ 0 \le x_{ij} \le c_{ij}$ ⌊Capacity Constraint

$\forall i \ne s,d \,;\ \sum_{j=1}^{n} x_{ij} - \sum_{K=1}^{n} x_{Ki} = 0$ ⌊flow conservation

Screen of LP formulation of the given graph

maxflow_model.lpt

```
Maximize
f_sw + f_sx + f_sz

Subject To

f_sw >= 0
f_sx >= 0
f_sz >= 0
f_wy >= 0
f_wt >= 0
f_xw >= 0
f_xy >= 0
f_xz >= 0
f_yt >= 0
f_zy >= 0
f_zt >= 0

f_sw <= 4
f_sx <= 7
f_sz <= 10
f_wy <= 2
f_wt <= 10
f_xw <= 2
f_xy <= 10
f_xz <= 2
f_yt <= 7
f_zy <= 2
f_zt <= 6

f_sw + f_xw - f_wy - f_wt = 0
f_sx - f_xw - f_xy - f_xz = 0
f_sz + f_xz - f_zy - f_zt = 0
f_wy + f_xy + f_zy - f_yt = 0

End
```

Output of Execution of LP Solve

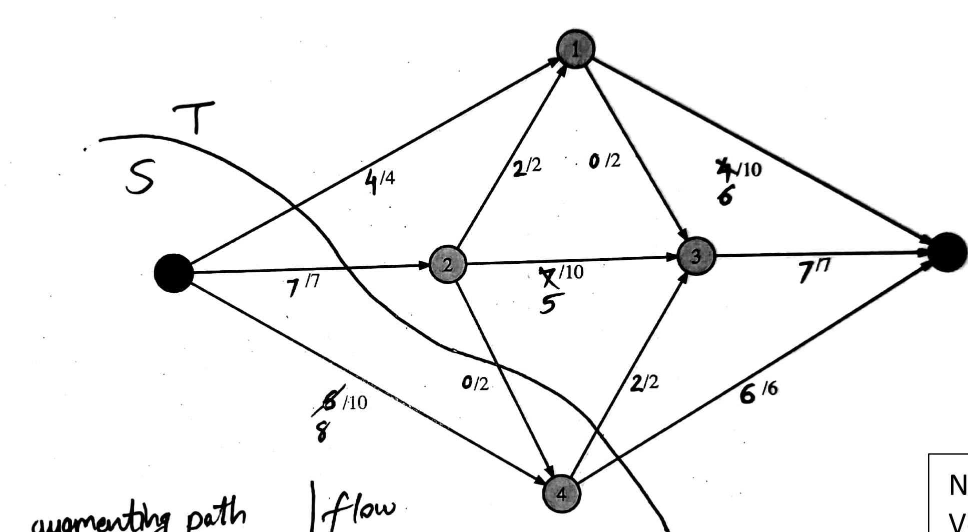
```
YPRADHAN-M-W3Z6:lp_solve_5.5.2.5_exe_osx32 ypradhan$ ./lp_solve -rxli xli_CPLEX maxflow_model.lpt
set_XLI: Successfully loaded 'xli_CPLEX'

Value of objective function: 19.00000000

Actual values of the variables:
f_sw                            4
f_sx                            7
f_sz                            8
f_wy                            0
f_wt                            6
f_xw                            2
f_xy                            5
f_xz                            0
f_yt                            7
f_zy                            2
f_zt                            6
YPRADHAN-M-W3Z6:lp_solve_5.5.2.5_exe_osx32 ypradhan$
```

The Value of the Max Flow is 19, and flow over the links are shown in the above figure.

Graphical Solution: Hand solving



augmenting path	flow
s - 1 - t : 4	4
s - 2 - 3 - t : 7	11
s - 4 - t : 6	17
s - 4 - 3 - 2 - 1 - t : 2	19 Max Flow

Min ST cut
S = {s, 4}
T = {1, 2, 3, t}

Max Flow : 19

Note: Mapping for Vertices

0	s
1	w
2	x
3	y
4	z
5	t

Solving it by running ford Fulkerson algorithm.

The source code resides at src/ypp170130, instructions to compile and run the code are given the src/readme file, the code has detailed commenting as well :)

Explanation of the Algorithm:

The algorithm has as input a flow network with edged having capacity, which indicates the maximum flow that can flow between vertices linked by corresponding edge. All vertices except the source and sink follow flow conservation law discussed in lp formulation.
We start by finding any initial flow that is flowing in the given network and accumulate to maxflow variable.

Next, we try to find and augmenting path from the source 0 to destination V-1. If we get an augmenting path, it means we can augment flow equivalent to bottleneck of the augmenting path, hence maxflow increases by value of bottleneck. Key thing to remember is augmenting path can consist of either non-full forward edges (i.e. flow is less than capacity) and non-empty backward edges (i.e. flow is positive). This augmenting path can be found in residual graph using any path finding technique, the current implementation uses Edmond Karp heuristic of using shortest path.

When we fail to find an augmenting path, it means that the flow is maximum that can go from source to destination given the network. In my implementation of the algorithm I have used Adjacency List representation of the graph which has better performance as compared to adjacency matrix representation.

Runtime: |V||E^2|

Output of Execution of Code:

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_201.jdk/Contents/Home/bin/java ...
Current Flow: 17

Results:

Vertices:   6
Edges:      11

0:  0--->1 (4/4);  0--->2 (7/7);  0--->4 (8/10);
1:  1--->3 (0/2);  1--->5 (6/10);
2:  2--->1 (2/2);  2--->3 (5/10);  2--->4 (0/2);
3:  3--->5 (7/7);
4:  4--->3 (2/2);  4--->5 (6/6);
5:

Max Flow: 19

Min Cut:
S: 0 4
T: 1 2 3 5


Process finished with exit code 0
```

Note: Mapping for Vertices

0	s
1	w
2	x
3	y
4	z
5	t

Now as we have results of LP and Code execution, we have answers to the Questions asked:

(a) What is the value of this flow?
The Value of the flow is 17 units.

(b) Is this a maximum s-t flow in this graph? If not, find a maximum s-t flow.
This is not a maximum s-t flow, the value of the maximum s-t flow is 19 units.

(c) Find a minimum s-t cut. (Specify which vertices belong to the sets of the cut.)
S = {s, z}
T = {w, x, y, t}

Comparison of these 2 results:

On comparing the two results, we obtain the same max flow value which is 19 from the LP solver as well as run of ford-fulkerson algorithm.

Yes, the answer obtained from LP solver and code execution are an integer solution.
As the integrality theorem states that, if all capacities are integers, then there exists a maximum flow and the flow on each edge is an integer.
This is also evident from results of lp solve and ford-fulkerson algorithm, we got a maxflow with integer value and flow for all edges is also integer.

References:
1. Class Notes
2. Max Flow Min Cut by Robert Sedgewick:
https://www.coursera.org/learn/algorithms-part2/home/week/3