FLOW NETWORKS

Lecture 5

Notes by Roman Vakhrushev
For CS-GY 6043 Design and Analysis of Algorithms II
10/11/2023

Contents

I Graphs 3

I.1 Definitions 3

I.2 Example 3

I.3 Representations 3

II Flow Network 4

II.1 Definition 4

II.2 Example 4

III Ford-Fulkerson Method 5

III.1 Maximum Flow Problem 5

III.2 Residual Network (Naïve version) 5

III.3 Outline of Ford-Fulkerson Method 6

III.4 Problem with This Definition of Residual Network 6

III.5 Residual Network (Final version) 7

IV Correctness of Ford-Fulkerson Method 9

IV.1 Useful Definitions, Facts, and Optimality condition 9

IV.2 Proof of Optimality of Ford-Fulkerson Method 10

IV.3 Example of Ford-Fulkerson Termination 12

IV.4 Termination and Running Time of FF 12

I Graphs

I.1 Definitions

Def: Directed graph $G = (V, E)$, where V is the set of vertices, E is the set of directed edges.

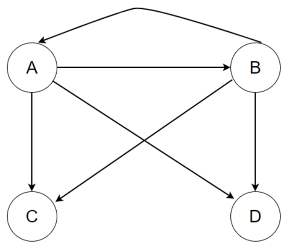
We use $|V| = n$ to denote number of vertices, $|E| = m$ to denote number of edges. Edges are ordered pairs of vertices. Both V and E are finite sets (infinite case is possible, but was not considered in the lecture).

I.2 Example

$G = (V, E)$

$V = \{A, B, C, D\}$

$E = \{(A, B), (A, C), (A, D), (B, C), (B, D)\}$



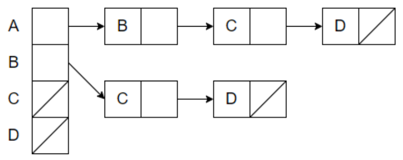
I.3 Representations

For the same graph as in example:

- Adjacency matrix:

$$\begin{pmatrix} \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 1 \\ B & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 0 \end{array} \end{pmatrix}$$

- Adjacency list:



II Flow Network

II.1 Definition

Def: Flow network is defined as (G, c, s, t) as follows:

$G = (V, E)$ - a directed graph

$s, t \in V$, where s - source (no incoming edges), t - sink (no outcoming edges).

Assumption 1:

$\forall v \in V$: $s \rightsquigarrow v \rightsquigarrow t$. (**Def:** $v \rightsquigarrow w$ means there is a directed path from v to w).

Assumption 2:

Between any two vertices, there is at most one edge.

Additionally, given capacity on edges:

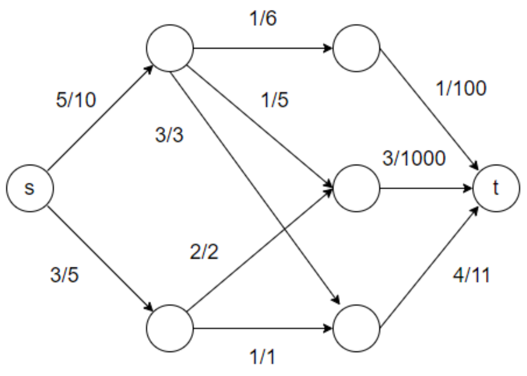
$c : E \to \mathbb{R}^+$ $[c(e) > 0, \forall e \in E]$

We define flow: $f : E \to \mathbb{R}_0^+$ with two properties:

1. **Capacity constraint:** $0 \leq f(e) \leq c(e)$
2. **Conservation of flow - "What goes in must come out":** for any $v \neq s, v \neq t$:

$$\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$$

II.2 Example



III Ford-Fulkerson Method

III.1 Maximum Flow Problem

Value of flow is defined as follows (s is a source):

$$|f| = \sum_{(s,v)\in E} f(s,v)$$

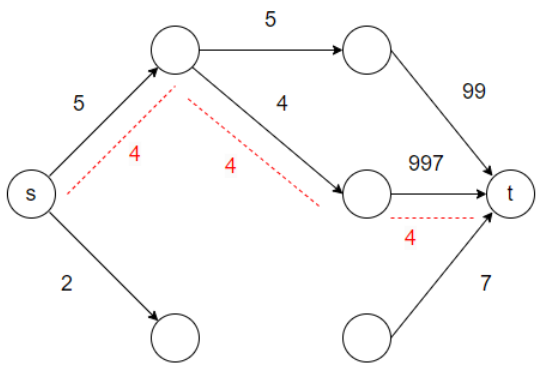
Given G and c, we want to find f maximizing $|f|$.

One way to solve maximum flow problem is by using Ford-Fulkerson Method (FF)

III.2 Residual Network (Naïve version)

Given a graph G with s, t, c, f, define residual network G_f : $c_f(u,v) = c(u,v) - f(u,v)$ if $f(u,v) < c(u,v)$.

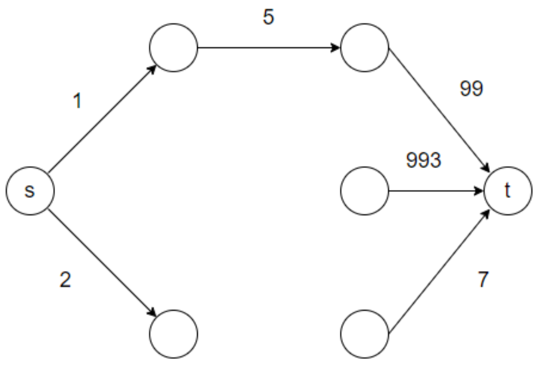
This is a residual network for our previous example. G_f :



The augmenting path is any path from s to t in residual network (one possible augmenting path is shown in red dashed line).

Claim: By increasing flow on the edges of the augmenting path (in this example, by min(5,4,997) = 4), we cannot obtain another valid flow.

If we increase the flow using the augmenting path as above, we end up with new flow network, which has the following residual network:



We often use f' to denote augmenting path (or more generally, flow), and $f \uparrow f'$ to denote updated flow.

III.3 Outline of Ford-Fulkerson Method

FF:

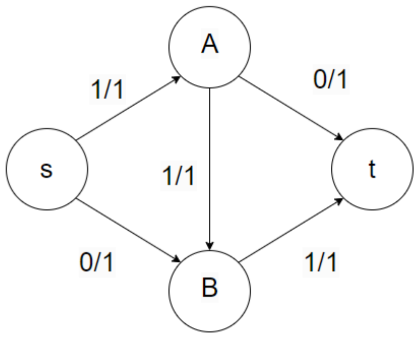
- Start with $f \equiv 0$
- Build G_f
- Find augmenting path
- Update f
- Repeat until no more augmenting paths

Sidenote 1: 0 is always a valid flow ($f \equiv 0$) by our definition of the flow.

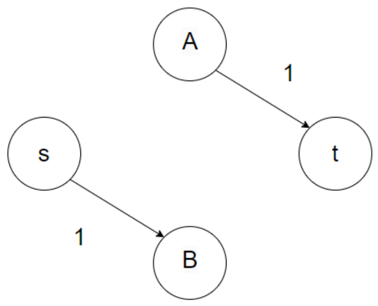
Sidenote 2: The way to find/choose augmenting path is not specified. Possible choices are BFS (also called Short Pipe), DFS, Fat Pipe (max improvement per update).

III.4 Problem with This Definition of Residual Network

If we define the residual network exactly the way we have done it in section III.2, we might get end up in situations, when FF does not find the maximum flow. Consider the following flow network:



Note that this flow is not maximum, since clearly there is a way to have flow value of 2 (instead of 1 in the flow network above). However, our residual network,G_f, looks as follows:



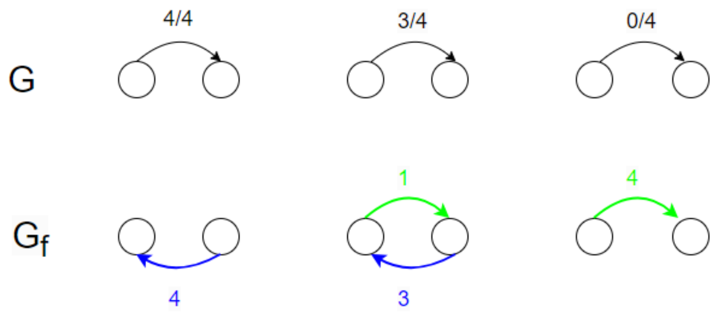
There is clearly no augmenting path, so FF method would terminate at this point without finding the maximum flow value.

III.5 Residual Network (Final version)

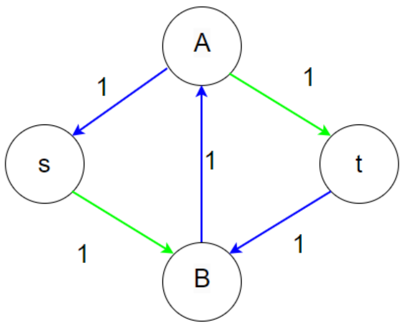
To overcome the issue above, we introduce the following modification to the definition of residual network we gave in section III.2.

New additional rule: $c_f(u, v) = f(v, u)$ if $f(v, u) > 0$.

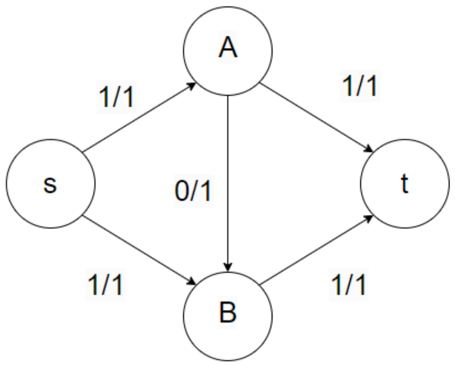
So, essentially, we allow reversing the existing flow when looking for an augmenting path on the residual networks. These edges are based on the flow in the flow network. Although it is not a standard convention, but it might be useful to use different colors to differentiate between the two types of edges. In this lecture green was used for original edges, blue was used for reversed edges . For example:



So, if we come back to our example given in section III.4, the residual network would look like this:



Note that in this residual network there is an augmenting path $s - B - A - t$, which allows us to improve the flow value. So, after one more round of FF method, we would end up with the following flow (which represents the maximum flow):



IV Correctness of Ford-Fulkerson Method

IV.1 Useful Definitions, Facts, and Optimality condition

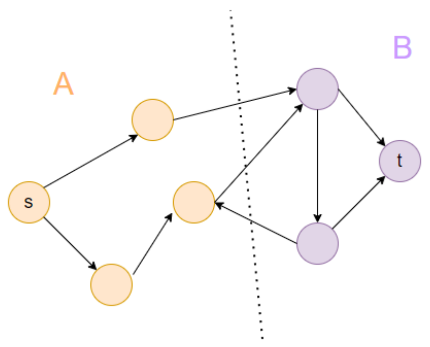
Def 1: $s - t$ cut in G:

$A, B \subset V$ and $A \cap B = \varnothing$, $A \cup B = V$, $s \not\subset A$, $t \in B$.

So, essentially, $s - t$ cut is a partition of vertices of a graph in two groups, such that s and t are in different groups.

Oftentimes it is useful to use sets A and B explicitly, so we use the specific version of $s - t$ cut, (A, B) $s - t$ cut.

A visual representation of a $s - t$ cut in some graph G:



Def 2: Flow across an (A, B) $s - t$ cut is defined as follows:

$$f(A,B) = \sum_{u \in A, v \in B, (v,u) \in E} f(u,v) - \sum_{u \in B, v \in A, (u,v) \in E} f(v,u)$$

Def 3: Capacity of (A, B) in some (A, B) $s - t$ cut is defined as follows:

$$c(A,B) = \sum_{u \in A, v \in B, (u,v) \in E} c(u,v)$$

Fact 1: For a fixed f: any (A, B) $s - t$ cut has the following property: $f(A, B) = |f|$.

Fact 2: For any f,c, (A, B) $s - t$ cut, the following is true: $f(A, B) \leq c(A, B)$.

By combining Fact 1 and Fact 2, we know the following is always true: $|f| = f(A, B) \leq c(A, B)$. This implies $\max_f |f| \leq \min_{(A,B)} c(A, B)$. But do we know when $\max_f |f| = \min_{(A,B)} c(A, B)$? Turns out, this statement is always true and is called the **max-flow min-cut theorem**.

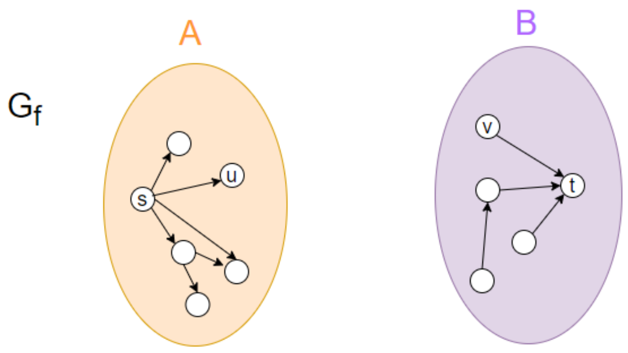
Thm: Max-flow min-cut theorem (Maximum flow - minimum (capacity) cut theorem):

$\max_f |f| = \min_{(A,B)} c(A, B)$, where value (A, B) is an $s - t$ cut.

IV.2 Proof of Optimality of Ford-Fulkerson Method

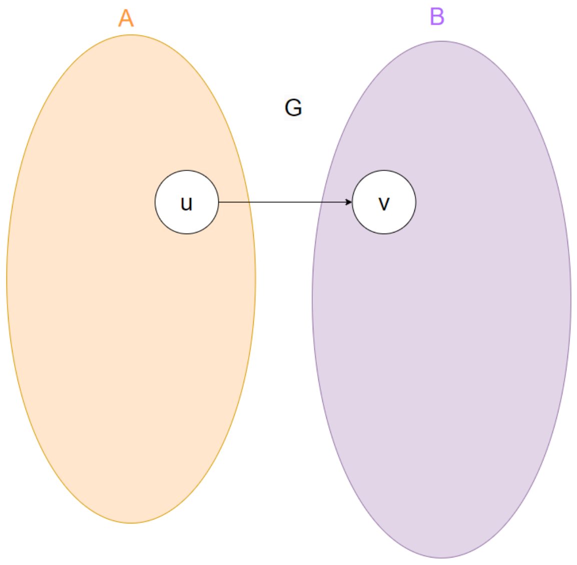
Claim: If FF stops, we can always find a (A, B) $s - t$ cut, such that $c(A, B) = |f|$.

Proof: Assume FF stops. Build G_f. Let A be a set of all vertices reachable from s in G_f. Let $B = V - A$. We are going to show that $c(A, B) = f(A, B) = |f|$.



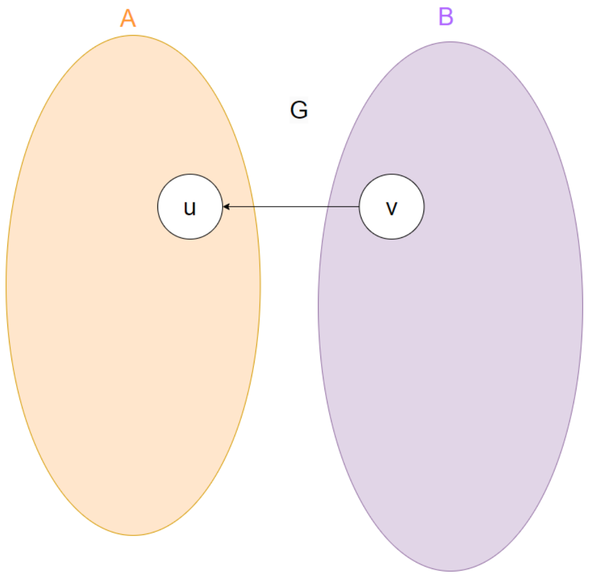
Consider any vertices u, v such that $u \in A$, $v \in B$ and there is an edge between them in graph G.

Case 1: $(u, v) \in E(G)$.



Then, clearly $f(u, v) = c(u, v)$. Otherwise, since u is reachable from s in G_f, so would be v, contradicting $v \in B$.

Case 2: $(v, u) \in E(G)$.



Then, $f(u,v) = 0$. Otherwise, there would be an edge in G_f (from u to v), since u is reachable from s in G_f, so would be v, contradicting $v \in B$.

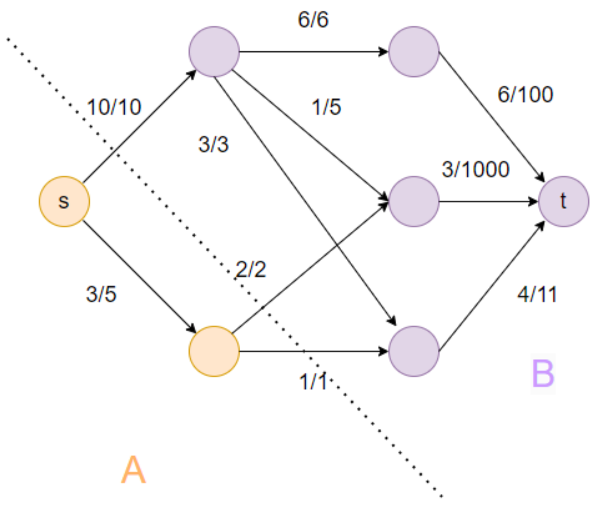
So, combining this information and definitions of $f(A,B)$ and $c(A,B)$, we have:

$$\begin{aligned}
f(A,B) &= \sum_{u\in A, v\in B, (v,u)\in E} f(u,v) - \sum_{u\in B, v\in A, (u,v)\in E} f(v,u) && (1)\\
&= \sum_{u\in A, v\in B, (v,u)\in E} f(u,v) && (2)\\
&= \sum_{u\in A, v\in B, (v,u)\in E} c(u,v) && (3)\\
&= c(A,B) && (4)
\end{aligned}$$

Thus, we have $f(A,B) = c(A,B)$. By using Fact 1 from section IV.1, we have $f(A,B) = c(A,B) = |f|$.

By max-flow min-cut theorem (or simply from inequality $\max_f |f| \leq \min_{(A,B)} c(A,B)$, which is trivial), we know we have found a maximum flow.

IV.3 Example of Ford-Fulkerson Termination



Recall our old example from section II and III. If we allow FF method to do a few more rounds, we will have the flow network above. In this network, it is clear, that if we have an $s - t$ cut exactly as shown in the picture, both flow and capacity of the cut have the value of 13. This implies we have obtained a maximum flow.

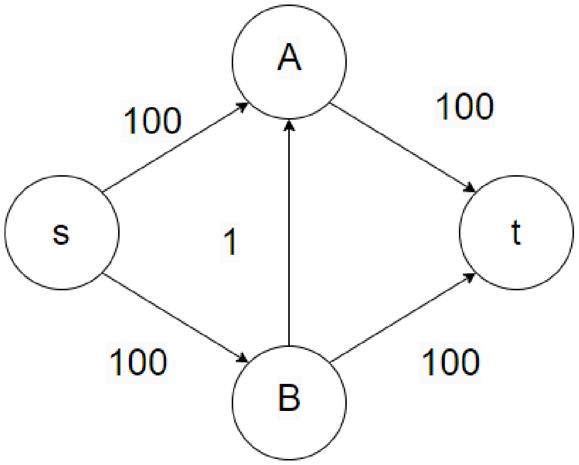
IV.4 Termination and Running Time of FF

In section IV.2 we have shown that FF method finds maximum flow **if** FF method terminates. However, in general, there is no guarantee that FF terminates. In fact, using some irrational capacity values, it is possible to construct examples of a flow network, on which FF will never terminate.

At the same time, we know that FF algorithm is guaranteed to terminate on integer capacity values.

Thm: If c is integral (integer-valued) function, then the FF method terminates. Additionally, the algorithm terminates after at most $|f^*|$ rounds, where $|f^*|$ is the maximum flow.

The running time of the algorithm partially depends on capacities, not on the size of network. This is unfortunate news, since it is possible to construct very small networks, which may take a lot of running time. For example, consider the following network (numbers are capacities):



This network is very small in terms of number of edges and vertices, however, if we always make a bad choice (i.e. if we choose augmenting paths with (A, B) or (B, A) edges every time), the FF method will run for 200 rounds.