CSS.201.1 Algorithms

Instructor: Umang Bhaskar

TIFR 2024, Aug-Dec

Scribe: Soham Chatterjee

sohamchatterjee999@gmail.com
Website: sohamch08.github.io

Contents

Chapter 1

Maximum Flow

1.1 Flow

Suppose we are given a directed graph $G = (V, E)$ with a source vertex s and a target vertex t. And additionally for every edge $e \in E$ we are given a number $c_e \in \mathbb{Z}_0$ which is called the capacity of the edge.

Definition 1.1.1: Flow

An $s - t$ flow is a function $f : E \longrightarrow \mathbb{R}_0$ which satisfies the following:

① $\forall\, e \in E,\, f(e) \le c_e$

② $\forall\, v \in V \setminus \{s, t\},\ \sum\limits_{e\in in(v)} f(e) = \sum\limits_{e\in out(v)} f(e)$

Also the value of a flow f is denoted by $|f| := \sum\limits_{e\in out(s)} f(e)$.

Before proceeding into the setup and the problem first we will assume some things

Assumption.

- *$in(s) = \emptyset$ i.e. there is no edge into s.*
- *$out(t) = \emptyset$ i.e. there is no edge out of t.*
- *There are no parallel edges*

Lemma 1.1.1

For any flow f, $|f| = \sum\limits_{e\in in(t)} f(e)$

Proof: We have for every edge $e \in E$, $\exists\, v \in V$ such that $e \in in(v)$ and $\exists\, u \in V$ such that $e \in out(u)$. Hence we get

$$\sum_{e\in E} f(e) = \sum_{v\in V}\sum_{e\in in(v)} f(e) = \sum_{v\in V}\sum)e \in out(v) f(e) \implies \sum_{v\in V}\left[\sum_{e\in in(v)} f(e) - \sum_{e\in out(v)} f(e)\right] = 0$$

Now we know $\forall\, v \in V \setminus \{s, t\}$. $\sum\limits_{e\in in(v)} f(e) = \sum\limits_{e\in out(v)} f(e)$. Therefore we get

$$\sum_{v\in V}\left[\sum_{e\in in(v)} f(e) - \sum_{e\in out(v)} f(e)\right] = 0 \implies \sum_{v\in\{s,t\}}\left[\sum_{e\in in(v)} f(e) - \sum_{e\in out(v)} f(e)\right] = 0 \implies \sum_{e\in out(s)} f(e) - \sum_{e\in in(t)} f(e)$$

Hence we have $|f| = \sum\limits_{e\in in(t)} f(e)$. ■

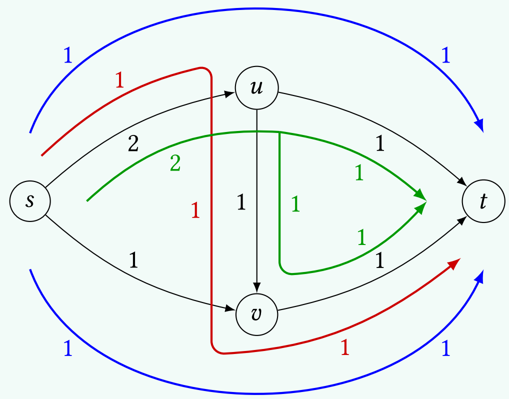
Max Flow

Input: A directed graph $G = (V, E)$ with source vertex s and target vertex t and for all edge $e \in E$ capacity of the edge $c_e \in \mathbb{Z}_+$

Question: Given such a graph and its capacities find an $s - t$ flow which has the maximum value

Example 1.1.1

Consider the following directed graph with capacities: $V = \{s, t, u, v\}$, $c_{s,u} = 2, c_{s,v} = c_{u,t} = c_{v,t} = c_{u,v} = 1$. Firstly the following function: $f' : f'(s,u) = 2 = f(u,t)$. It is not a flow since $f(u,t) = 2 > 1 = c_{u,t}$. Now we define three different flow functions:



- f: $f(s,u) = f(u,v) = f(v,t) = 1$ and otherwise 0. Therefore $|f| = 1$
- g: $g(s,u) = g(u,t) = 1$, $g(s,v) = g(v,t) = 1$ and otherwise 0. Therefore $|g| = 2$
- h: $h(s,u) = 2$, $h(u,t) = h(u,v) = h(v,t) = 1$ and otherwise 0. Therefore $|h| = 2$

Notice here g and h has the maximum flow value.

1.2 Ford Fulkerson Algorithm

Definition 1.2.1: Residual Graph

Given a directed graph $G = (V, E)$ and capacities C_e for all $e \in E$ and an $s - t$ flow f the residual graph $G_f = (V, E_f)$ has the edges with the following properties:

1. If $(u,v) \in E$ and $f(u,v) > 0$ then $(v,u) \in E_f$ and $c^f_{v,u} = f(u,v)$. Such an edge is called a "backward" edge.
2. If $(u,v) \in E$ and $f(u,v) < c_{u,v}$ then $(u,v) \in E_f$ and $c^f_{u,v} = c_{u,v} - f(u,v)$. It is called "forward" edge.

Algorithm 1: Max-Flow

Input: Directed graph $G = (V, E)$, source s, target t and edge capacities C_e for all $e \in E$

Output: Flow f with maximum value

```
1 begin
2     for e ∈ E do
3         f(e) = 0
4     while ∃ s ⇝ t path P in G_f do
5         δ ⟵ min_{e∈P}{c_e^f} for e = (u,v) ∈ P do
6             if e is Forward Edge then
7                 f(u,v) ⟵ f(u,v) + δ
8             else
9                 f(u,v) ⟵ f(v,u) − δ
```

We call one iteration of the While loop at line 4 *Flow Augmentation*.

Lemma 1.2.1
At any iteration the f' obtained after the flow augmentation of the flow f is a valid flow

Proof: At any iteration let P be the path from $s \rightsquigarrow t$ and $\delta = \min_{e \in P} c_f(e)$. Let f' be the new function such that for each $(u,v) \in P$ if (u,v) is forward edge in G_f then $f'(u,v) = f(u,v) + \delta$ and if (u,v) is backward edge in G_f then $f'(v,u) = f(v,u) - \delta$ and for other edges $e \in E \setminus P$, $f'(e) = f(e)$.

Now since $\delta = \min_{e \in P} c_f(e)$, $c_f(e) \geq \delta$ for all $e \in P$. Hence if (u,v) is backward edge then $(v,u) \in E$ and $c_f(u,v) = f(u,v)$. Hence $f'(v,u) = f(v,u) - \delta \geq 0$. Therefore for all $e \in E$, $f'(e) \geq 0$.

Now first we will show $f'(e) \leq c_e$ for all $e \in E$. If $(u,v) \in P$ is a forward edge then $(u,v) \in E$ and $c_f(u,v) = c_{u,v} f(u,v)$. Therefore $f'(u,v) = f(u,v) + \delta \leq f(u,v) + c_{u,v} - f(u,v) = c_{u,v}$. Now if $(u,v) \in P$ is a backward edge then $(v,u) \in E$ and $c_f(u,v) = f(u,v)$. Therefore $f'(u,v) = f(u,v) - \delta \leq f(u,v) \leq c_{u,v}$. For other edges $e \in E \setminus P$, $f'(e) = f(u) \leq c_e$. Therefore $f'(e) \leq c_e$ for all $e \in E$

Now we will prove for all $v \in V \setminus \{s,t\}$, $\sum_{e \in in(v)} f'(e) = \sum_{e \in out(v)} f'(e)$. If v is not in the path P in G_f then, $f'(e) = f(e)$ for all edges $e \in in(v) \cup out(v)$. Hence the condition is satisfied for such vertices. Suppose v is in the path P. Then there are two edges e_1 and e_2 in P which are incident on e. If both are forward edges or both are backward edges then one of them is in $in(v)$ and other one is in $out(v)$. WLOG suppose $e_1 \in in(v)$ and $e_2 \in out(v)$ we have

$$\sum_{e \in in(v)} f'(e) = \sum_{e \in in(v) \setminus \{e_1\}} f(e) + f(e_1) \pm \delta = \sum_{e \in out(v) \setminus \{e_2\}} f(e) + f(e_2) \pm \delta = \sum_{e \in out(v)} f'(e)$$

If one of e_1, e_2 forward edge and other one is backward edge then either $e_1, e_2 \in in(v)$ (when e_1 is forward and e_2 is backward) or $e_1, e_2 \in out(v)$ (when e_1 is backward and e_2 is forward). Now if $e_1, e_2 \in in(v)$, $f'(e_1) + f'(e_2) = f(e_1) + \delta + f(e_2) - \delta = f(e_1) + f(e_2)$ and if $e_1, e_2 \in out(v)$ then $f'(e_1) + f'(e_2) = f(e_1) - \delta + f'(e_2) + \delta = f(e_1) + f(e_2)$. Hence

$$\sum_{e \in in(v)} f'(e) = \sum_{e \in in(v)} f(e) = \sum_{e \in out(v)} f(e) = \sum_{e \in out(v)} f'(e)$$

Hence f' is a valid flow. ■

Lemma 1.2.2
At any iteration Given G_f if the flow, f' obtained after flow augmentation of f by δ then

$$|f'| = |f| + \delta$$

Proof: Since we augment flow along an $s \rightsquigarrow t$ path, the first edge of the path is always in $out(s)$. Let the first edge is $e = (s,u)$. Now e has to be a forward edge because otherwise $(u,s) \in E$ and then there is an incoming edge in G which is not possible. Hence

$$|f'| = \sum_{e \in out(s)} f'(e) = \sum_{e \in out(s) \setminus \{e\}} f(e) + f'(e) = \sum_{e \in out(s) \setminus \{e\}} f(e) + f(e) + \delta = \sum_{e \in out(s)} f(e) + \delta = |f| + \delta$$

Hence we have the lemma. ■

Lemma 1.2.3 A
every iteration of the Ford-Fulkerson Algorithm the flow values and the residual capacities of the residual graph are non-negative integers.

Proof: Initial flow and the residual capacities are non-negative integers. Let till i^{th} iteration the flow values and the residual capacities were non-negative integers. Let the flow after i^{th} iteration was f. Hence $\forall\, e \in E$, $f(e) \in \mathbb{Z}_0$. Therefore in the G_f for all $e \in E_f$, $c_f(e) \in \mathbb{Z}_0$. Hence $\delta \in \mathbb{Z}_0$. Therefore $\forall\, e \in E$, $f'(e) \in \mathbb{Z}_0$. And therefore for all $e \in E_{f'}$ where $G_{f'}$ is the residual graph of the flow f', $c_{f'}(e) \in \mathbb{Z}_0$. Hence by mathematical induction the lemma follows. ■

At any iteration let P be the path from $s \rightsquigarrow t$. Then for all $e \in P$, $c_f(e) > 0$. Therefore $\delta = \min\limits_{e \in P} c_f(e) \geq 1$. Therefore the algorithm must stop in at most $\sum\limits_{e \in out(s)} c_e$ since we can have the value of a flow to be at max the value of the sum of capacities of edges in $out(s)$ and therefore we can increase the flow at max that many times.

Lemma 1.2.4 |

f is a max flow then there is no $s \rightsquigarrow t$ path in G_f.

Proof: Suppose there is an $s \rightsquigarrow t$ path P in G_f. We will show that then f is not a max flow following the algorithm. Then $\forall\ e \in P$, $c_f(e) > 0$. Hence $\delta = \min\limits_{e \in P} c_f(e) \geq 1$. Now after the flow augmentation process of f by δ we get a new valid flow f' by Lemma 1.2.1 and by Lemma 1.2.2 we have $|f'| = |f| + \delta > ||f|$. Hence f is not a maximum flow. Hence contradiction. Therefore there is no $s \rightsquigarrow t$ path in G_f. ■

1.3 Cuts

Definition 1.3.1: Cut Set

For a graph $G = (V, E)$ and a subset $A \subseteq V$, the cut $(A, V \setminus A)$ is a bipartition of V where the edges E_A of the graph $G_A = (A, V \setminus A, E_A)$ is the set $E_A = E \cap (A \times (V \setminus A))$.

Now if s, t are two vertices of G then an $s - t$ *Cut* $(A, V \setminus A)$ is a cut such that $s \in A$ and $t \in V \setminus A$.

Now we define for a cut $(A, V \setminus A)$ the *Capacity of the Cut* $(A, V \setminus A) = \sum\limits_{e \in E_A} c_e$.