Max Flow

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

The Max Flow Problem

$G = (V, E)$: a directed simple graph.

Given a vertex $v \in V$, we will use

- $\text{IN}(v)$ to denote the set of in-neighbors of v;
- $\text{OUT}(v)$ to denote the set of out-neighbors of v.

G has no "isolated" vertices v with $\text{IN}(v) = \text{OUT}(v) = \emptyset$.

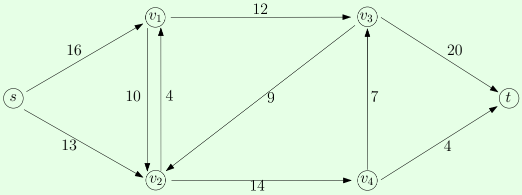
s: a vertex in G called the **source vertex**, which satisfies $\text{IN}(s) = \emptyset$
t: a vertex in G called the **sink vertex**, which satisfies $\text{OUT}(t) = \emptyset$

$\mathbb{N}$: The set of integers
$\mathbb{N}_{\geq 0}$: The set of non-negative integers

We are given a capacity function $c : V \times V \to \mathbb{N}_{\geq 0}$ satisfying:

- $c(u, v) = 0$ if $(u, v) \notin E$.

Example:



The numbers indicate the edge capacities.

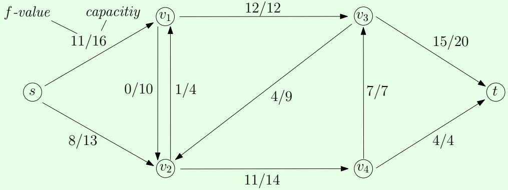
The Max Flow Problem

A **flow** is function $f : V \times V \to \mathbb{N}_{\geq 0}$ satisfying:

- $f(u, v) \leq c(u, v)$ for any $u \in V$ and $v \in V$;
- for each vertex $v \in V \setminus \{s, t\}$

$$\sum_{u \in \text{IN}(v)} f(u, v) = \sum_{w \in \text{OUT}(v)} f(v, w).$$

Example:



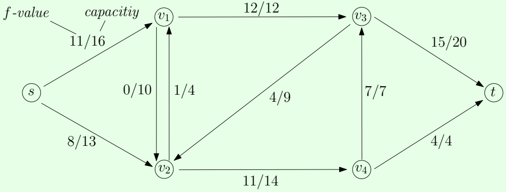
For each edge, we represent its f-value x and c-value y in the form of x/y.

The Max Flow Problem

Given a flow f, we define its **value** as:

$$|f| = \sum_{u \in \mathrm{OUT}(s)} f(s, u).$$

Example:

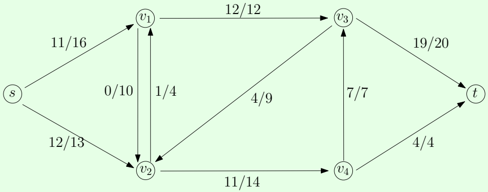


This flow has value 19.

The Max Flow Problem

Goal of the Max Flow Problem:
Find a flow f with the maximum value $|f|$.

Example:



A max flow with value 23.

Lemma 1: For any flow f, it holds that

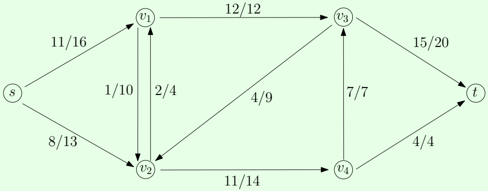
$$|f| = \sum_{u \in \mathrm{IN}(t)} f(u, t).$$

The proof is left as an exercise.

A flow f is **clean** if it has the property that:
for any $u \in V$ and $v \in V$, if $f(u, v) > 0$, then $f(v, u) = 0$.

Otherwise, the flow is **dirty**.

Example: A dirty flow:



The lemma below suggests that we can focus on clean flows.

Lemma 2: The maximum value of all flows equals the maximum value of all clean flows.

The proof is left as an exercise.

Next, we will introduce the notion of "general flow", which is essential for understanding the mathematical structure of the max-flow problem. There is a one-one correspondence between clean flows and general flows.

A **general flow** is a function $F : V \times V \to \mathbb{N}$ satisfying:

- $F(u, v) \leq c(u, v)$;
- $F(u, v) = -F(v, u)$;
- for any $u \in V \setminus \{s, t\}$, it holds that

$$\sum_{v \in V} F(u, v) = 0.$$

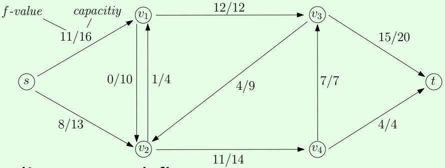
The **value** of F is defined as

$$|F| = \sum_{u \in \text{OUT}(s)} F(s, u).$$

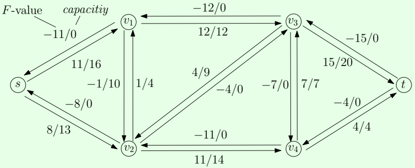
Lemma 3: For every clean flow f, there is a general flow F with $|F| = |f|$. Conversely, for every general flow F, there is a clean flow f with $|f| = |F|$.

The proof is left as an exercise.

Example: A clean flow



The corresponding general flow



Henceforth, we will focus on finding a general flow F with the maximum value $|F|$.

Given a general flow F, we can define the **residual function** $R : V \times V \to \mathbb{N}_{\geq 0}$ by setting

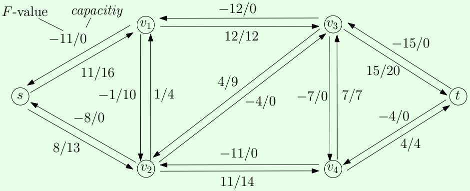
$$R(u, v) = c(u, v) - F(u, v)$$

for any $u \in U$ and $v \in V$.

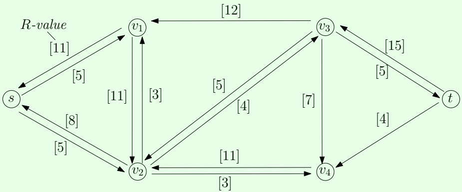
We can now define the **residual graph** as $G_F = (V, E_F)$ where $E_F = \{(u, v) \in V \times V \mid R(u, v) > 0\}$.

Example:

A general flow F:

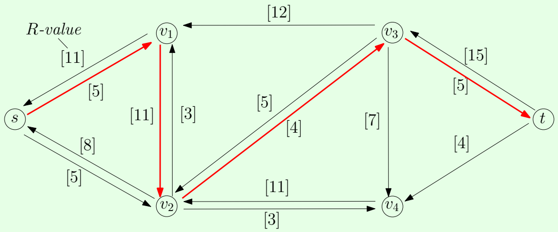


The residual graph G_F:



An **augmenting path** is a simple path π from s to t in G_F.
The **capacity** of π is the smallest R-value of the edges on π.

Example:
The red path is an augmenting path with capacity 4.



Given G_F, we can find an augmenting π in $O(|E|)$ time.

How to do so with BFS/DFS?

We will show how to use an augmenting path to obtain a better flow. Identify an arbitrary augmenting path π, and let Δ be the capacity of π.

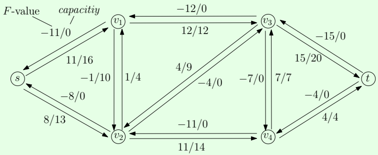
Define a function $F' : V \times V \to \mathbb{N}$ as follows. For each $u \in V$ and $v \in V$

- if (u, v) is on π, then $F'(u, v) = F(u, v) + \Delta$;
- if (v, u) is on π, then $F'(u, v) = F(u, v) - \Delta$;
- otherwise, $F'(u, v) = F(u, v)$.

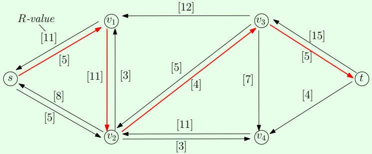
Lemma 5: The function F' thus constructed is a general flow with $|F'| = |F| + \Delta$.

The proof is left as an exercise.

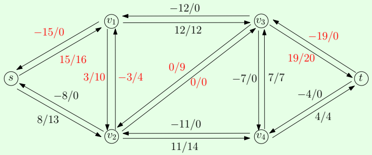
Example: A general flow F:



An augmenting path π in the residual graph G_F:



The new general flow F':



The Ford-Fulkerson Algorithm

1. $F \leftarrow$ a general flow mapping every edge to 0
2. **do**
3. $\quad G_F \leftarrow$ the residual graph decided by F
4. $\quad$ find an augmenting path π
5. $\quad$ **if** π does not exist **then** return F
6. $\quad F' \leftarrow$ the new general flow constructed from F and π
7. $\quad F \leftarrow F'$

The rest of the slides will prove

- the algorithm is correct;
- its running time is $O(|E|M)$, where M is the maximum flow value.

Let S and T be subsets of V satisfying

- $S \cup T = V$ and $S \cap T = \emptyset$;
- $s \in S$ and $t \in T$.

We say that they make an **S-T cut**.

Lemma 6: $|F| = \sum_{u \in S, v \in T} F(u, v)$ holds for any general flow F and any S-T cut.

The proof is left as an exercise.

We define the **capacity** of an S-T cut as

$$c[S, T] \quad = \quad \sum_{u \in S, v \in T} c(u, v).$$

The following can be obtained immediately from the previous lemma.

Corollary 7: $|F| \leq c[S, t]$ holds for any general flow F and any S-T cut.

The Max-Flow Min-Cut Theorem:
Let F be an arbitrary general flow. The following statements are equivalent:

1. F has the largest value among all general flows.
2. G_F (the residual graph decided by F) has no augmenting paths.
3. There exists an S-T cut such that $|F| = c[S, T]$.

Proof. We will prove

- statement 1 $\Rightarrow$ statement 2
- statement 2 $\Rightarrow$ statement 3
- statement 3 $\Rightarrow$ statement 1

which imply that all three statements are equivalent.

Proof of $1 \Rightarrow 2$**.** If F has the maximum value, then G_F has not augmenting path. Otherwise, the Ford-Fulkerson algorithm yields a general flow with a strictly larger value.

Proof of $3 \Rightarrow 1$**.** If F does not have the maximum value, then let F' be a general flow with $|F'| > |F|$. But then $|F'| > |F| = c[S, T]$, contradicting Corollary 7.

Proof of $2 \Rightarrow 3$. Define

$$\begin{aligned} S &= \{v \in V \mid S \text{ can reach } v \text{ in } G_F\} \\ T &= V \setminus S. \end{aligned}$$

Clearly, $s \in S$. As G_F has no augmenting paths, we know $t \notin T$.

Consider any $u \in S$ and any $v \in T$. The edge (u, v) cannot exist in G_F (otherwise, s would be able to reach v, contradicting $v \in T$).

Let R be the residual function decided by F. The fact that (u, v) does not exist in G_F tells us $R(u, v) = 0$. Hence,
$R(u, v) = c(u, v) - F(u, v) = 0$, implying $c(u, v) = F(u, v)$.

It thus follows that

$$\begin{aligned} |F| &= \sum_{u \in S, v \in T} F(u, v) \qquad \text{(by Lemma 6)} \\ &= \sum_{u \in S, v \in T} c(u, v) \\ &= c[S, T]. \end{aligned}$$

□

The correctness of the Ford-Fulkerson algorithm follows from the max-flow min-cut theorem.

As a corollary, the maximum flow value M must be an integer (the proof left as an exercise).

Each time the algorithm finds an augmenting path, the value of our flow increases by **at least** 1. Therefore, the algorithm performs at most $M + 1$ iterations. The total running time is therefore $O(|E| \cdot M)$.

Remark: If we use the BFS algorithm to find an augmenting path in each iteration, it is possible to show that the Ford-Fulkerson algorithm finishes in $O(|V||E|^2)$ time. The details are not required in this course.