20 Max-flow/Min-cut, Bipartite Matching

- Max flow/min cut
- Bipartite matching

20.1 Max-flow/Min-cut

For any $S \subseteq V$, let $f^{in}(S) = \sum_{e \text{ into } S} f(e)$, $f^{out}(S) = \sum_{e \text{ out of } S} f(e)$.

Lemma 20.1
Let f be any $s-t$ flow, and let (A, B) be any $s-t$ cut.
Then $v(f) = f^{out}(A) - f^{in}(A)$.

Proof. By definition, $v(f) = f^{out}(s) = f^{out}(s) - f^{in}(s)$. $(f^{in}(s) = 0)$

For any internal vertex v, $f^{out}(v) - f^{in}(v) = 0$. So,

$$v(f) = \sum_{v \in A}(f^{out}(v) - f^{in}(v))$$

There are three cases that we have to consider.

- e has both ends in A: appears in both $+$ and $-$ terms.
- e goes out of A: appears only in $+$ term.
- e goes into A: appears only in $-$ term.

Thus, $v(f) = f^{out}(A) - f^{in}(A)$. □

Lemma 20.2
Let f be any $s-t$ flow, and let (A, B) be any $s-t$ cut.

Then $v(f) \leq c(A, B)$.

Proof.

$$\begin{aligned} v(f) &= f^{out}(A) - f^{in}(A) \\ &\leq f^{out}(A) \\ &= \sum_{e \text{ out of } A} f(e) \\ &\leq \sum_{e \text{ out of } A} c_e \\ &= c(A, B) \end{aligned}$$

□

Theorem 20.3 (Max-flow/Min-cut Theorem)
For any flow network, the largest value of any flow equals the minimum capacity of any cut.

Proof. Let f be a flow such that there is no $s-t$ path in its residual flow network.
Let A be the set of vertices v with an $s-v$ path in the residual network of f.
Let $B = V \setminus A$.

We claim that $v(f) = c(A, B)$.

Then the theorem will follow, since by the lemma, the flow with the maximum value will always be less than or equal to the cut with the minimum capacity.

Why is (A, B) an $s-t$ cut?
Clearly, $s \in A$. there is no $s-t$ path in the residual network, so $t \in B$.

How do the flows relate to the capacities?

- Consider an edge $e = (u, v)$ with $u \in A$, $v \in B$.

  If $f(e) < c_e$, then e is a forward edge in the residual network, so the path from s to u could be extended to an $s-v$ path, which is a contradiction. Therefore, $f(e) = c_e$.

- Consider an edge $e' = (u', v')$ where $u' \in B$, $v' \in A$.

  If $f(e') > 0$, then (v', u') is a backward edge in the residual network. So the path from s to v' could be extended to an $s-u'$ path, which is again a contradiction. Therefore, $f(e') = 0$.

Thus,

$$\begin{aligned} v(f) &= f^{out}(A) - f^{in}(A) \\ &= \sum_{e \text{ out of } A} c_e - \sum_{e \text{ into } A} 0 \\ &= c(A, B) \end{aligned}$$

□

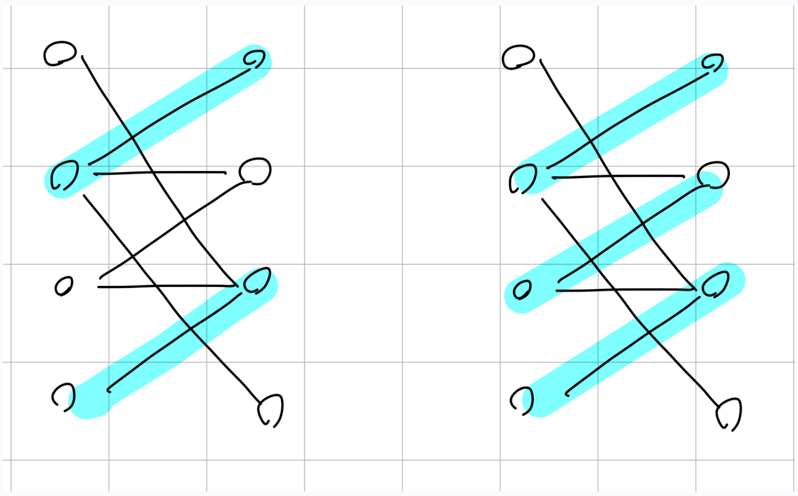
20.2 Maximum Bipartite Matching

Recall that a bipartite graph $G = (V, E)$ has $V = A \cup B$, $A \cap B = \varnothing$ such that all edges have one end in A and one end in B.

Definition 20.4
A **matching** in G is an edge subset $M \subseteq E$ such that each vertex appears in at most one edge of M.

The problem that we want to solve, is that given a bipartite graph, we want to find a matching that includes as many edges as possible.

Example 20.5
Take the following bipartite graph:



These are two valid matchings, the maximum number of edges we could get is 3.