CS217 – Algorithm Design and Analysis
Homework 5

Not Strong Enough

May 8, 2020

1: Exercise 1

Let $G = (V, c)$ be a flow network. Prove that flow is "transitive" in the following sense: if r, s, t are vertices, and there is an r-s-flow of value k and an s-t-flow of value k, then there is an r-t-flow of value k.

Proof. Note that there is an r-s-flow of value k means that the value of the maximum r-s-flow is at least k, which also means that the value of the minimum r-s-cut is at least k. Similarly, the value of the minimum s-t-cut is also at least k.

Now consider an r-t-cut. It is either an r-s-cut (if s is not in the cut) or an s-t-cut (if s is in the cut). So the capacity of the minimum r-t-cut is at least k. It follows that the value of the maximum r-t-flow is at least k, and thus there is an r-t-flow of value k. $\square$

2: Exercise 3

Prove Menger's Theorem. You have to prove two things: first, not both cases above can occur (this is rather easy); second, one of them must occur (this requires a tool from the lecture).

Proof. Let $V(G) = \{v_1, \cdots\}$ and $s = v_p, t = v_q$.

We would like to construct a flow network (V', s', t', c) where $V' = \{v_1, v_1', v_2, v_2', \cdots\}$ and

$$c(u,v) = \begin{cases} 1, \text{ if } \exists i, (u,v) = (v_i, v_i') \\ \infty, \text{ if } \exists i,j, (u,v) = (v_i', v_j) \wedge (v_i, v_j) \in E \\ 0, \text{ otherwise} \end{cases}.$$

And finally let $s' = v_p', t' = v_q$. Then:

- There are k vertex disjoint paths $p_1, \cdots, p_k$ in G iff there is a flow f in (V', s', t', c) with $\mathrm{val}(f) = k$, iff $\max \mathrm{val}(f) \geq k$.
- There are $k-1$ vertices $v_{i_1}, \cdots, v_{ik-1}$ in $V \setminus \{s,t\}$ such that $G - \{v_{i_1}, \cdots, v_{i_{k-1}}\}$ contains no $s-t$ path iff there is a cut S in (V', s', t', c) with $\mathrm{cap}(S) = k-1$ by making $v_{i_j} \in S$ and $v_{i_j}' \notin S$ for all j, iff $\min \mathrm{cap}(S) < k$.

By *Max-Flow Min-Cut Theorem*, let $\max \mathrm{val}(f) = \min \mathrm{cap}(S) = l$. Then:

- Either $l \geq k$, resulting in 1 holds while 2 does not.
- Or $l < k$, resulting in 2 holds while 1 does not.

Therefore, *exactly one* of the two statements is true. $\square$

In the two exercises below, let $\Gamma(X)$ be the neighbors of X.

3: Exercise 4

Consider the induced bipartite subgraph $H_n[L_i \cup L_{i+1}]$, show that for $i < n/2$ the graph has a matching of size $|L_i| = \binom{n}{i}$

Proof. Use Hall's Theorem, the size of maximum matching equals $\min_{X \subseteq L_i} |L_i| - |X| + |\Gamma(X)|$.

Since in $H_n[L_i \cup L_{i+1}]$ the degree of each vertex in L_i is $n - i$, and that of each vertex in L_{i+1} is $i + 1$, there is $|X|(n - i) \leq |\Gamma(X)|(i + 1)$. As $i < n/2$, $|X| \leq |\Gamma(x)|\frac{i+1}{n-i} \leq |\Gamma(X)|$, and only if $|X| = 0$ can the equality be achieved.

So there is $\min_{X \subseteq L_i}(|L_i| - |X| + |\Gamma(X)|) = |L_i| = \binom{n}{i}$. $\square$

4: Exercise 5

Show that there are $\binom{n}{i}$ paths in H_n starting at L_i ending in L_{n-i} and are disjoint.

Proof. We first set a new point s connected to all vertices in L_i and a new point t connected to all vertices in L_{n-i}.

Set the capacity of edges starting from s or ending to t be 1, and the capacity of the rest be ∞. Then there is a flow that, for edges whose capacity is 1 the flow takes 1; for the rest edges, each in form of $(u, v), u \in L_k, v \in L_{k+1}$, the flow takes $\frac{\binom{n}{i}}{\binom{n}{k}(n-k)}$.

It is obvious that the flow is well-defined, that for each vertex (except for s, t) the flow in equals the flow out. And the total flow is $\binom{n}{i}$. Besides, it is the maxflow since the flow out of s is no more than $\binom{n}{i}$.

So there exists a min-cut of the graph that the size of the cut is $\binom{n}{i}$ as well. The collection of the end point of each edge in the cut is a vertex cut, so the size of the minimum vertex cut is $\binom{n}{i}$. By Menger's Theorem, there are $\binom{n}{i}$ disjoint paths from s to t, by removing s and t of these paths, we get $\binom{n}{i}$ disjoint paths from L_i to L_{n-i}

□

5: Exercise 6

Let $\nu(G)$ denote the size of a maximum matching of $G = (V, E)$. Show that a bipartite graph G has at most $2^{\nu(G)}$ mimimum vertex covers.

Proof. From the König's Theorem, we know that the size of mimimum vertex cover is $\nu(G)$ if G is a bipartite graph. Let C be a minimum vertex cover. Then we can construct new minimum vertex covers by choosing vertices from C and $V \setminus C$. In other words, all minimum vertex covers can be represented by $X \cup Y$, where $X \subseteq C, Y \subseteq V \setminus C$. Denote $N(A) = \{b \mid \exists a \in A, \text{ there is an edge between } a \text{ and } b\}$. For all $X \subseteq C$, to construct a vertex cover, Y must touch all edges touched by $C \setminus X$ but not by X.

- If $N(C \setminus X) \cap (C \setminus X) \neq \emptyset$, there does not exist such a Y.
- If $N(C \setminus X) \cap (C \setminus X) = \emptyset$, then Y must be at least $N(C \setminus X) \cap (V \setminus C)$ to be a vertex cover. To make $X \cap Y$ a minimum vertex cover, Y has to be $N(C \setminus X) \cap (V \setminus C)$.

Since X has $2^{\nu(G)}$ choices, G has at most $2^{\nu(G)}$ mimimum vertex covers. $\square$

Obviously, this is not true for general (non-bipartite) graphs: the triangle K_3 has $\nu(K_3) = 1$ but it has three minimum vertex covers. The five-cycle C_5 has $\nu(C_5) = 2$ but has five minimum vertex covers.

6: Exercise 7

Is there a function $f\colon \mathbf{N_0} \to \mathbf{N_0}$ such that every graph with $\nu(G) = k$ has at most $f(k)$ minimum vertex covers? How small a function f can you obtain?

Solution. Suppose that we have a graph $G = (V, E)$ and one of its maximum matching $M \subseteq E$ with $|M| = \nu(G) = k$.

We have the two following observations:

- For any vertex cover $C \subseteq V$ of G, for every edge in M there must be at least one of its endpoint which is in C. Otherwise there exists an edge in M such that neither of its endpoints is in C, which means that this edge is uncovered and therefore C is not a vertex cover.
- For any vertex v which is not matched, all of its neighbors must be matched, or the edge between v and one of its unmatched neighbors can be added to the maximum matching and therefore M is not maximum.

We now construct a vertex set $C_0 \subseteq V$ such that for every edge (u, v) in M, either $u \in C_0$, or $v \in C_0$, or both $u, v \in C_0$. There are 3^k possible C_0 in total.

For each possible C_0, note that it may not be a "vertex cover" by far. So we try to construct another vertex set C_1 from C_0. Let C_1 be an empty set at the beginning. From the second observation above, for every unmatched vertex v, there are two cases. If all of its neighbors are in C_0, then we do nothing, since every edge connected to v is covered by vertices in C_0. Otherwise we add it into C_1 to cover the edges which C_0 didn't cover. Note that C_1 is *uniquely determined by* C_0.

Let $\mathcal{C}$ be a family of vertex covers, which is initialized to empty. Now consider $C_0 \cup C_1$. We know that it is also uniquely determined by C_0. If it is a vertex cover, we add it to $\mathcal{C}$. There are at most 3^k vertex covers in $\mathcal{C}$, since there are 3^k possible C_0, and for some C_0 and its corresponding C_1, $C_0 \cup C_1$ may not be a vertex cover.

Claim that any minimum vertex cover C must belong to $\mathcal{C}$. Because from the first observation, we can let the unique C_0 be the matched vertices covered by C. And then unique C_1 can be constructed from C_0. $C_0 \cup C_1$ is the minimum vertex cover when C_0 is fixed. So $C = C_0 \cup C_1 \in \mathcal{C}$.

So there are at most 3^k minimum vertex covers in total, and $f(k) = 3^k$. Also note that this upper bound is *tight*. Just consider the triangle K_3 — it has $3 = 3^1 = 3^{\nu(K_3)}$ minimum vertex covers. $\square$