Tutorial 3: Dynamic Programming

Problem 1. Let $G = (L \cup R, E)$ be a bipartite graph. A cycle cover is a collection of vertex disjoint cycles $C_1, \ldots, C_k$ such that every vertex of G belongs to exactly one cycle. Design an algorithm that finds a cycle cover for G.

Hint: Convert G to an s-t flow network G' and find a maximum flow in G'.

Solution. We assume that a single edge counts as a cycle. We construct G' as follows:

- For every vertex $u \in V$ create two vertices u_0 and u_1 in G'.
- For every edge $(u, v) \in E$ of G, add edges (u_0, v_1) and (v_0, u_1) to G'.
- Add a source s and a sink t and add the edges $(s, u_0), (u_1, t)$ for every $u \in V$.
- Give all edges capacity 1.

Claim. *There is a cycle cover of G if and only if there is a flow in G of value $|V|$.*

Proof. Given a cycle cover $\mathcal{C}$ of G, we construct a flow f in G' as follows: We set $f(s, u_0) := 1$ and $f(u_1, t) := 1$ for every $u \in V$, and for each cycle $(u_1, \ldots, u_k)$ in $\mathcal{C}$, we set

$$f(u_{10}, u_{21}) := 1, f(u_{20}, u_{31}) := 1, \ldots, f(u_{k0}, u_{10}) := 1$$

and $f(e) := 0$ for every other edge.

Conversely, suppose that there is a flow in G' of value $|V|$. Take an integral flow f of the same value. Since all edges have capacity 1, it must be that for every edge e, $f(e)$ is either 0 or 1, and that $f(s, u_0) = 1$ and $f(u_1, t) = 1$ for all $u \in V$. To construct a cycle cover of G, pick a vertex u_1 of G. Since $f(s, u_{10}) = 1$, there must be an edge (u_{10}, u_{21}) in G', hence there must be an edge (u_1, u_2) in G. Similarly, since $f(s, u_{20}) = 1$, there must be an edge $(u_{2,0}, u_{3,0})$ in G' and an edge (u_2, u_3) in G. If $u_3 = u_1$, then (u_1, u_2) is a cycle. Otherwise, we continue, until we find an edge (u_{k-1}, u_k) with $u_k = u_1$. $(u_1, \ldots, u_k)$ will be a cycle in G. If it does not contain all vertices, we start again with a vertex not in $(u_1, \ldots, u_k)$ to find a second cycle, and so on. Notice that these cycles will be vertex disjoint. $\square$

Problem 2. Consider an undirected graph $G = (V, E)$ and two vertices $s, t \in V$. A set X is an s-t vertex separator if every path between s and t visits some vertex in X and $s, t \notin X$ (in other words, if s and t become disconnected when we remove X from G). Design an algorithm that computes a minimum vertex separator in G.

Hint: Convert G to a (directed) flow network $G' = (V', E')$ as follows:

- For every vertex $u \in V$ create two vertices u_0 and u_1 in G'.
- For every u, add an edge (u_0, u_1) to G'.
- For every edge $(u, v) \in E$ of G, add edges (v_1, u_0) and (u_1, v_0) to G'.
- Assign all edges appropriately chosen capacities.

Is the assumption that G is undirected important?

Solution. We give capacity 1 to edges of the form (u_0, u_1) and infinite capacity to all other edges. We show that vertex separators in G correspond to cuts in G':

Claim. *There is a vertex separator of size s in G if and only if there is a cut of size s in G'.*

Proof. Suppose X is a vertex separator of size s in G, and let S and T two sets of vertices in $G - X$ such that $s \in S$, $t \in T$, and there is no path from S to T in $G - X$. (S', T') where S' consists of all vertices u_0, u_1 for $u \in S$ and all vertices u_0 for $u \in X$, and $T' = V(G') - S'$, is a cut of size s in G'.

Conversely, a cut of size s in G' will necessarily consist entirely of edges of the form (u_0, u_1) since these are the only edges with finite capacity. Then the set X of all vertices u for these edges will form a vertex separator of size s in G. $\square$

In particular, minimum cuts in G' correspond to minimum vertex separators in G. If G was directed, we would change item 3 of the construction to only include edges (u_1, v_0) in G' for every edge (u, v) of G.