1 Completeness Problems

In this lecture we will continue looking at a variety of computational problems, and analyzing them to determine which class each problem falls under. A lot of these problems can be solved by deriving from other problems which have already been solved or of which we have already looked at. The three we looked at last lecture include SAT, Independent Set and Subset Sum. This lecture picks up where the last one left off by first looking at the clique problem.

1.1 Clique Problem

This problem is defined by the following question: given a graph $\{< G, K > \mid G$ has a clique of size $\geq k\}$. A clique is defined as a subset of all vertices in a graph G, all of which are adjacent to each other, where clique is a complete subgraph of of G. We can prove that this problem is in NP via a fairly trivial proof. To prove it is in NP-complete, we will show that it can be reduced to the independent set problem.

Proof: We are going to prove this by showing that INDPSET $\leq_p$ CLIQUE. Suppose that G is the input to INDPSET. Then let $\overline{G}$ be the complement of G. Now suppose that G has an independent set S of size k. Then, S is a clique in $\overline{G}$. The same will hold in the other direction. Thus, we see that the CLIQUE problem can be reduced to the INDPSET problem. ■

We can see that this reduction can be done in polynomial time because we simply remove edges and then add edges to our graph where there are no edges. Thus is takes linear time to convert our instance of G into $\overline{G}$.

1.2 Vertex Cover Problem

The vertex cover problem is defined by the following question: given a graph $\{< G, K > \mid G$ has a vertex cover of size $\leq k\}$. Proving that VERTEX COVER is in NP time is also fairly trivial, and can essentially be proven in one line. Given a set that we claim is vertex cover, we just check every edge in the overall graph and make sure that one of its endpoint is in that set. The proof certificate is just the subset of edges. As we know, to prove a problem is NP-complete, we first must prove that the problem is NP-hard, which will be our first course of action here. We will use an existing NP-complete problem, and then reduce that problem to our problem, which is VERTEX COVER in this case. We can use INDPSET again for this reduction. Just to review, for the vertex cover problem, we are wanting to find a subset of vertices such that all the edges are incident.

Proof: We are going to prove this by showing that INDPSET $\leq_p$ VERTEX COVER. Suppose that G has an independent set S of size s. Then every edge in G has at least one endpoint in $V \backslash S$. Then $V \backslash S$ is a vertex cover of G that is of size $(n - s)$. The other direction will hold the same and is similar to prove. ■

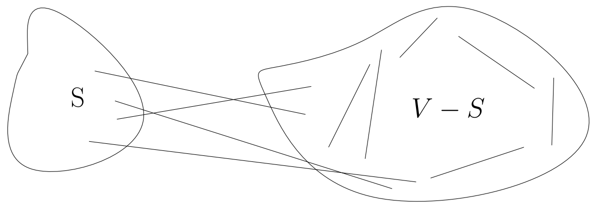

Fig. 1: Vertex Cover Proof Example

1.3 Edge Cover Problem

Edge cover is defined as a subset of edges $M \subseteq E$ such that for all vertices in the graph ($v \in V$), there exists an edge in M ($e \in M$) such that a vertex is incident to this edge ($v \in e$). When looking at this problem, we first must ask, is edge cover in NP? We can verify that this problem is in polynomial time by giving an edge covering, then checking it's size to see if it is less then or equal to k. Now we must see if this problem is NP-hard, so that we can then verify if it is NP-complete. But in this case, we do have a polynomial time algorithm to find the smallest edge cover. This can be done by find a maximum matching and then extending it greedily so that all of the vertices are covered. Thus, this problem is not NP-hard, or NP-complete for that matter.

The edge cover is a great example of a problem that looks very similar to a problem that is NP-complete (vertex cover), but is actually quite easy to verify a solution in polynomial time.

1.4 Hamiltonian Cycle (Hamcycle)

Below is the Peterson Graph; it is a famous type of graph used to give lots of counterexamples and examples with computer theory.

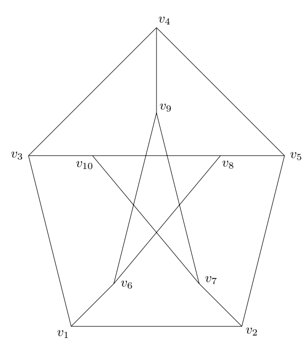

Fig. 2: Petersen Graph

Remark 1.1 *Is the Petersen graph Hamiltonian?*

A Hamiltonian Cycle is defined as a graph where $\{G \mid G \text{ } is \text{ } Hamiltonian\}$. In order to prove that a graph, such as the Petersen graph, does not have a Hamiltonian circuit, there is no general way or short answer on how to do this. In general, you have to look at all possible cycles and verify that none of them work. We can verify that this problem is in NP by counting the vertices and then checking that each vertex is connected to the next one by an edge, and that the last vertex is connected to the first vertex. It takes time proportional to n, since there are n number of vertices to count and n edges to check. Now that we know this problem in is NP, we must prove that it is in NP-complete.

Theorem 1.1 *A HAMCYCLE is in NP-Complete.*

Proof: We knows that HAMCYCLE is in NP. Given a graph G with a cycle, we can verify in linear time whether the cycle is Hamiltonian. To prove that HAMCYCLE is NP-hard, we will show that VERTEX COVER $\leq_p$ HAMCYCLE. Let G be the input of VERTEX COVER, and we will construct an instance G' of HAMCYCLE. For every edge (u, v) in G, create a edge-gadget in G' (this can be seen below in Figure 3). Given this structure, there is only three possible ways to traverse all of the the vertices, given a 'u-side' and a 'v-side', as pictured below:

1. Enter from the u-side, go somewhere else in the graph, and then come back through the other side (v-side)
2. Enter and exit through the u-side
3. Enter and exit through the v-side

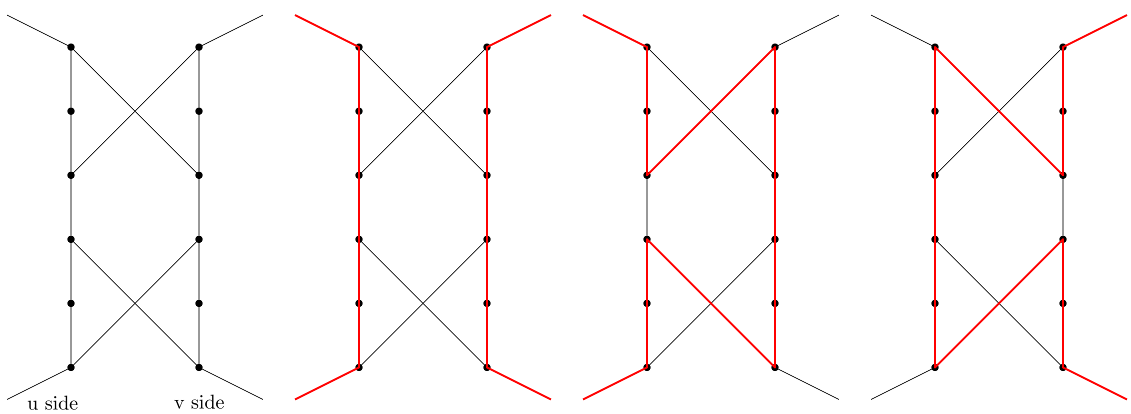

Fig. 3: Edge-Gadgets with a u side and v side

Suppose we have a graph with 5 vertices, such as the one in figure 4. For every edge, we will create a gadget such that there is a positive and negative end of each line (this can be seen in figure 4). We then label each side of each gadget such that it corresponds to the appropriate vertex in our graph. We will connect each gadget by going from a negative end to a positive end. We can use these gadgets to cover the same number of vertices as we would if we were actually traversing the actual graph. Next, we will add k number of selector vertices (s_1, s_2 to s_k in this case). For every u, we will connect the first positive u and the last negative u to each selector.

If we have a vertex cover $(\mu_1, ..., \mu_k)$ of size k, the idea would be to use paths to cover all of the edges that are covered by each item in the vertex cover, iterating through the list one by one. To

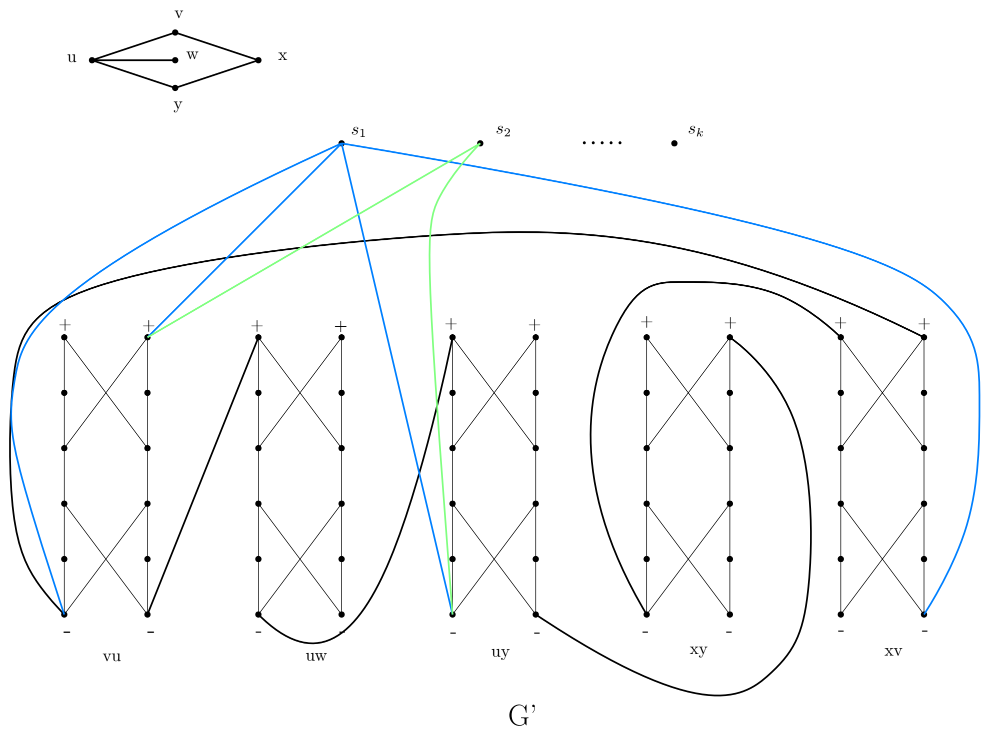

Fig. 4: Edge-Gadget Graph of G'

start, we will connect the selector vertices by attaching edges between them and the structures we have so far. This concept is a bit confusing, but basically each vertex will be connected to each selector variable at both the positive and negative ends.

Continuing with our proof, we will show that the size of G' is $polynomial(|G|)$. Suppose that G has n-vertices and m-edges. Then we know the number of vertices in $G' = 12n + k \leq 13n$ and that the number of edges in $G' = 14m + (2m - n) + 2kn = O(m + n^2)$. Thus, this reduction can be done in polynomial time.

The final and crucial part of this proof is to show that if the original graph has a vertex cover of size k, then this new graph has a Hamiltonian cycle. We will use the same graph as before, but this time just adding one edge from u to x, in order to show all of the possible cases (this graph has the same cover, size = 2). We will show that this construction has a Hamiltonian cycle, starting from s_1. The key to remember here is that if both endpoints of an edge are in the cover, then we will traverse those two parts independently (either enter/exit from the u-side or v-side). However if an edge is only covered by one vertex in the cover, then we will enter from the u-side and exit from the v-side. The final path is shown below with arrows showing it's direction. We can see in this figure that we have a path that starts from s_1 and ends from s_1, thus we have a Hamiltonian cycle. The same can be seen with s_2.

So in summary, if a graph has a vertex cover of size k, then this construction gives us a Hamiltonian cycle. We can create multiple G' up to k, and one of those graphs will have a Hamiltonian cycle. Thus we have proven that HAMCYCLE is NP-complete. ■

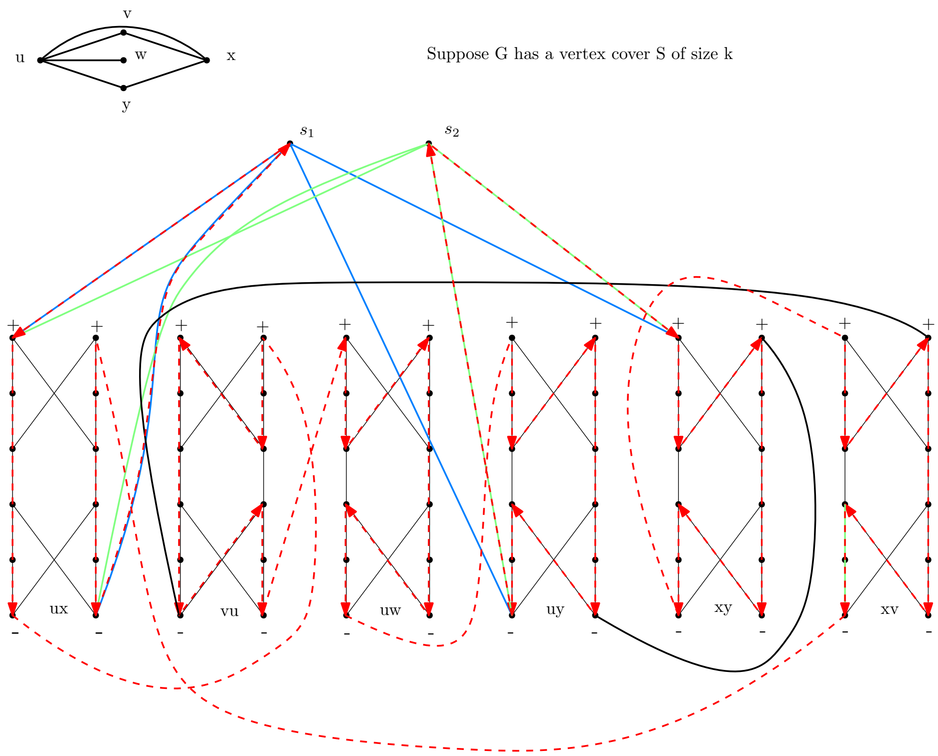

Fig. 5: Edge-Gadget Graph of G' with Flow

1.5 Traveling Salesman Problem (TSP)

This problem looks at the situation where given a graph $= \{G, cost, k \mid G$ has a Hamiltonian cycle of cost $\leq k\}$. TSP is trivial to show that it belongs to NP because if someone gives you a particular cycle you can verify if it is Hamiltonian cycle and the cost is $\leq k$ by simply counting up the cost. To show TSP is NP-complete, we can show that HAMCYCLE $\leq_p$ TSP.

Proof:

Start by creating a graph G' as the following: G' is complete, and $|V(G')| = |V(G)| = n$. The $cost(uv) = 0$ if $uv \in E(G)$. Else, then the $cost(uv) = 1$. G is Hamiltonian iff G' has a zero cost tour.

■