Algorithmic Lower Bounds - Assignment 2

Professor: Mohammad T. Hajiaghayi

Problem 1. An ϵ-imperfect 2-coloring is a 2-coloring of an undirected graph in which no more than an ϵ fraction of the edges have endpoints of the same color. Show that it is NP-complete to find an ϵ-imperfect 2-coloring for some constant ϵ strictly between 0 and 1 (i.e., ϵ cannot have any dependence on the input but can be whatever constant you want other than 0 or 1).

Solution. We reduce from Monotone NAE-SAT where every clause has exactly 3 variables, all 3 of which are unique. We let ϵ equal $\frac{1}{6}$. We represent each clause by three nodes connected in a triangle. Thus if all three nodes are the same color we have three monochromatic edges; however, if only two nodes are the same color we will only have one monochromatic edge. Next, we create a node for every variable and connect it to every node representing an instance of that variable. If any of the literals are the same color as the variable it creates a monochromatic edge. The number of edges in this graph is exactly six times the number of clauses. The graph must contain at least one monochromatic edge in each clause. Thus the graph is only $\frac{1}{6}$-imperfect 2-colorable if no other edges are monochromatic. This can only occur if all variable nodes are the opposite color of their literals, and thus all the literal are the same color, and if all the clauses have two different colors in them. We pick one color to be true and one to be false, when translating to NAE-SAT.

Problem 2. Give an approximation-preserving reduction from max cut to unique coverage.

Solution. Let $G = (V, E)$ denote an instance of the Maximum Cut problem. We now construct an instance $\mathcal{I} = (\mathcal{S},\mathcal{U})$ of unique coverage as follows - Set $\mathcal{S} = V$ and $\mathcal{U} = E$. To represent $(\mathcal{I},\mathcal{U})$ as a bipartite graph you would have the V on the left , E on the right, and a an edge from u to e if e has u as an endpoint.

Claim: G has a max cut of size $k \rightarrow \mathcal{I}$ has a unique coverage of size k.

The proof is simple. Let S denote the max cut in G, i.e. $|\delta(S, V \setminus S)| = k$. The same set $S \subset \mathcal{S}$ is now a set in $\mathcal{I}$ that uniquely covers k elements. This is because each edge in G is an element in $\mathcal{I}$ and all edges in the cut are uniquely covered.

Claim: $\mathcal{I}$ has a unique coverage of size $k \rightarrow G$ has a max cut of size k.

The proof is same as above. Let $S \subset \mathcal{S}$ denote the set that uniquely covers k elements. It is easy to observe that the same set of vertices $S \subset V$ has a cut of size k.

Problem 3. An *apex* graph is a graph that can be made planar by the removal of a single vertex. Given a graph $G = (V, E)$, we say that G has a *k-strong coloring* if vertices of G can be colored by at most k colors such that no two vertices sharing the same edge have the same color and every vertex in the graph dominates[1] an entire color class. Show that it is NP-hard to find a k-strong coloring in apex graphs for any $k \geq 4$.

Solution. We first prove the statement for the general graphs by giving a reduction from the k-coloring problem. Given an instance of the k-coloring problem in a graph G, we add a new vertex v and connect it to all vertices in G. If G has a k-coloring, then the new graph has a $(k+1)$-strong coloring by just coloring the new vertex with a new color. This coloring is proper and every vertex dominates the new color. Now suppose the new graph has a $(k+1)$-strong coloring. Since vertex v is connected to all other vertices, it should have a unique color and removing v results in a proper coloring of G with k colors.

Now we prove the statement for apex graphs. We know that the 3-coloring problem in planar graphs is NP-hard. By the definition of apex graphs, if we add a new vertex to a planar graph, the resulting graph would be an apex. Therefore, our reduction shows that 4-strong coloring in apex graphs is NP-hard. By a similar reduction, we show that k-strong coloring in apex graphs for $k > 4$ is also NP-hard. If we attach $(k-4)$ disjoint paths of length 2 to the vertex v, by a similar argument, it can be shown that the new graph has a k-strong coloring if and only if the planar graph G is 3-colorable.

[1] A vertex v dominates $\{v\} \cup N(v)$ where $N(v)$ is the set of neighbors of vertex v.

Problem 4. Given a graph $G = (V, E)$, we say the graph G is *beautiful* if we can color the vertices of G with either blue or red such that each vertex has **exactly one** blue neighbor. Either show that the problem of deciding G is beautiful is NP-hard, or show that there exists a polynomial-time algorithm for the problem.

Solution. We prove the NP-hardness of the problem by giving a reduction from the exact set cover problem. Given an instance of exact set cover with sets $\mathcal{S} = \{S_1, S_2, \cdots\}$ and elements $\mathcal{E} = \{e_1, e_2, \cdots\}$, we create a bipartite graph $G = (A, B, E)$ as follows. We add a vertex in A for each element in $\mathcal{E}$, and add a vertex in B for each set in $\mathcal{S}$. For each set $S \in \mathcal{S}$ and element $e \in S$, we add an edge between their corresponding vertices in G. We also attach a path of length 2 to the corresponding vertex of each set in $\mathcal{S}$. We show that graph G is beautiful if and only if the exact set cover has a solution.

First, if this graph is beautiful, then the blue vertices in $\mathcal{S}$ are a solution for the exact set cover problem because each element in $\mathcal{E}$ has exactly one blue neighbor in $\mathcal{S}$.

Now suppose exact set cover has a solution. We show that the graph G would be beautiful. Specifically, we show that it is possible to set the colors of the solution sets to blue. For a set S in our solution, we set its color to blue, and we color the first vertex in the path attached to S with blue and the second vertex in the path with red. Also, for a set S' which is not in the solution, we set its color to red, and we color two vertices in the path attached to it with blue. We color all other vertices with red. Clearly, every vertex has exactly one blue neighbor. This completes the reduction, so the problem is NP-hard.