6.892: Algorithmic Lower Bounds, Spring 2019

Prof. Erik Demaine, Jeffrey Bosboom, Jayson Lynch

Problem Set 1 Solution

Due: Tuesday, February 12, 2019 at noon

Problem 1.1 [Hamiltonian Cycle Problem → Traveling Salesman Problem].

Recall from lecture that a *Hamiltonian cycle* in a graph is a cycle visiting every vertex exactly once. The Hamiltonian Cycle Problem is the NP-complete problem of deciding whether the input graph has a Hamiltonian cycle.

The Traveling Salesman Problem in Graphs is to decide, given a complete graph with *positive integer* edge weights and given a target integer t, does there exist a cycle[1] in the graph that visits every vertex at least once having total weight $\leq t$?

Prove that the Traveling Salesman Problem in Graphs is NP-hard by reducing from the Hamiltonian Cycle Problem.

Solution: (This first solution is intentionally more verbose than we'd normally write, and than we'd expect of you, so that you have a clear example of a fully formal reduction proof. In your solutions, make sure you don't omit any of the steps of the proof, but feel free to prove them more briefly.)

We reduce from the Hamiltonian Cycle Problem to the Traveling Salesman Problem in Graphs. Given an instance of Hamiltonian Cycle, namely a graph $G = (V, E)$ with $n = |V|$ vertices, our reduction outputs the following instance of TSP in Graphs:

- A complete graph G' on the same vertex set V, where each edge is labeled with weight 1 if it is in E, and with weight $n + 1$ otherwise.
- The target weight $t = n$.

Having described the reduction, we need to prove two properties about it. First, we need to show that our reduction is polynomial time. It takes $O(n)$ time to count the number n of vertices and $O(n^2)$ time to create the weighted complete graph G', so our reduction is certainly polynomial time. Second, we need to show that our reduction is correct—that is, the resulting TSP in Graphs instance (G', n) always has the same answer as the original Hamiltonian Cycle instance G. We will do this by showing that a solution to either instance implies the existence of a solution to the other instance (in both directions).

In one direction, suppose that there exists a solution to the Hamiltonian Cycle instance, that is, that G has a Hamiltonian cycle C. We can view C as a cycle in G' as well. Because C is Hamiltonian, it visits all n vertices exactly once. Thus C has length n and so it has weight n in G', as it uses only edges in E. Cycle C visits every vertex at least once and has weight $\leq n$, so it is also a solution to the TSP in Graphs instance.

In the other direction, suppose that there exists a solution to the TSP in Graphs instance, that is, a cycle C in G' visits every vertex at least once and has weight $\leq n$. Because C visits every

[1] Throughout this class, a *cycle* is allowed to repeat vertices and/or edges; if it doesn't repeat vertices, the cycle is called *simple*.

vertex at least once, it has length $\geq n$.[2] On the other hand, C has weight $\leq n$ and thus length $\leq n$ (because all weights are positive integers). So the cycle C must have length exactly n, which means it visits every vertex exactly once. Finally, C must be a cycle in G because, if it included any edges not in E, then that edge alone would cause it to have weight at least $n+1$ (because weights are nonnegative), a contradiction. Therefore, C is a Hamiltonian cycle on G, so it is also a solution to the Hamiltonian Cycle Problem instance.

We have demonstrated a correct polynomial-time reduction from Hamiltonian Cycle to TSP in Graphs. Because the Hamiltonian Cycle Problem is NP-hard, so is the Traveling Salesman Problem in Graphs.

[2] Recall that the *length* of a cycle is the number of edges in the cycle (counting repetitions), which is equal to the number of vertices in the cycle (counting repetitions).