CPSC 320 Notes: The Futility of Laying Pipe, Part 2

March 19, 2017

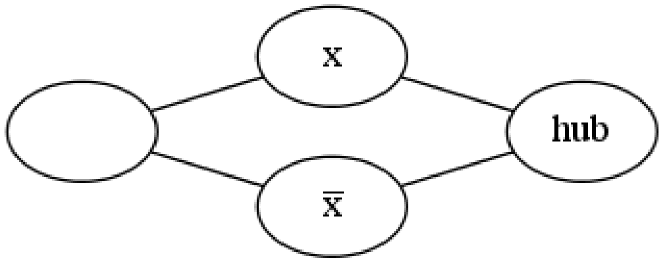
Reminder: An instance of the Steiner Problem (SP) is an undirected graph $G = (V, E)$ and a subset $S \subseteq V$ of the vertices to which we must deliver water. A solution to the instance is a subset $E' \subseteq E$ of the edges which connects all vertices in S (and perhaps some in V). The **best** solution is the one with the fewest edges. In the decision-variant, we add a variable k to an instance and ask whether a solution with at most k edges exists.

1 SP is NP-Complete

We've already shown that SP $\in$ NP. To show it's NP-complete, we need to also show that it is NP-hard; that is, that it's at least as hard as every other problem in NP. We'll use 3-SAT to help us, since we already know that 3-SAT is at least as hard as every other problem in NP.

1. Which of these would show that SP is at least as hard as 3-SAT: reducing from SP to 3-SAT in polynomial time or reducing from 3-SAT to SP in polynomial time? (Hint: Checking whether a list of n numbers is in sorted order is a simple problem that can be solved in polynomial time, which we'll call SORTED. There is a (silly, trivial) polynomial-time reduction one direction or the other between SORTED and 3-SAT. Whichever direction works easily for that is the **wrong** direction, since SORTED is only NP-complete if P=NP.)

2. Here is a sketch of a "variable gadget" to help with our reduction. How can we "shade in" (put in S) some of these vertices and choose an appropriate k (maximum number of edges in the solution) to enforce 3-SAT's choice that x or $\overline{x}$ is true but not both?



3. Draw a graph with four variable gadgets (x_1, x_2, x_3, and x_4), all sharing a single "hub" node. (Be sure your layout still enforces choosing either true or false but not both for each of the four variables, yet allows all 16 possible combinations of their truth values.)

4. Now, find a way to add one or more nodes and edges to your graph and choose a k in order to represent the clause $(\overline{x_1} \vee \overline{x_2} \vee x_3)$ and enforce that: at least one of $\overline{x_1}$, $\overline{x_2}$, and x_3 is true and also (still) each variable is either true or false but not both. (x_4 isn't in this clause, which is fine. In most 3-SAT problems, not all variables are in all clauses.)

5. Give a complete reduction from 3-SAT to SP such that the answer to the SP instance you produce is `YES` if and only if the answer to the original 3-SAT instance is `YES`.

6. Analyse the runtime of your reduction (to show that it takes polynomial time).