CS41 Lab 13

December 2 2019

This week, we'll continue exploring NP-COMPLETE decision problems, and develop approximation algorithms for related versions of those problems.

1. **The hardness of Three-Coloring-OPT**

   Recall the THREE-COLORING problem: Given a graph $G = (V, E)$, output YES iff the vertices in G can be colored using only three colors such that the endpoints of any edge have different colors. We know that THREE-COLORING is NP-COMPLETE. But what about the optimization version of THREE-COLORING?

   Let THREE-COLORING-OPT be the following problem. Given a graph $G = (V, E)$ as input, color the vertices in G using at most three colors in a way that maximizes the number of *satisfied* edges, where an edge $e = (u, v)$ is satisfied if u and v have different colors.

   Show that if there is a polynomial-time algorithm for THREE-COLORING-OPT then P = NP.

2. **Traveling Salesman Problem.** In this problem, a salesman travels the country making sales pitches. The salesman must visit n cities and then return to her home city, all while doing so as cheaply as possible.

   The input is a complete graph $G = (V, E)$ along with nonnegative edge costs $\{c_e : e \in E\}$. A *tour* is a simple cycle $(v_{j_1}, \ldots, v_{j_n}, v_{j_1})$ that visits every vertex exactly once.[1] The goal is to output the minimum-cost tour.

   For many TSP applications (such as when the cost is proportional to the distance between two cities), it makes sense for the edges to obey the *triangle inequality*: for every i, j, k, we have

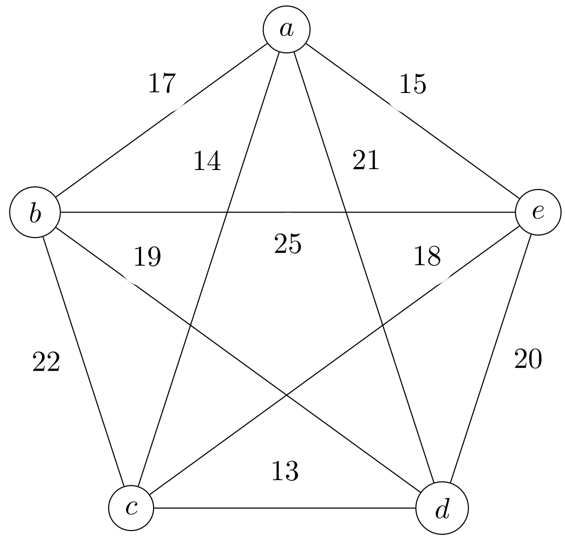
   $$c_{(ik)} \leq c_{(ij)} + c_{(jk)}.$$

   This version is often called METRIC-TSP.

   The (decision version of the) Traveling Salesman Problem is NP-COMPLETE. For this problem, you will develop a 2-approximation algorithm for METRIC-TSP.

   (a) First, to gain some intuition, consider the following graph:

[1] except for the start vertex, which we visit again to complete the cycle



(b) *On your own* try to identify a cheap tour of the graph.

(c) Build some more intuition by computing the minimum spanning tree (MST) of the graph. Let T be your minimum spanning tree.

(d) Let OPT be the cheapest tour. Show that its cost is bounded below by the cost of the MST: $\text{cost}(T) \leq \text{cost}(OPT)$.

(e) Give an algorithm which returns a tour A which costs at most twice the cost of the MST: $\text{cost}(A) \leq 2\,\text{cost}(T)$.

(f) Conclude that your algorithm is a 2-approximation for METRIC-TSP.

3. **Optimization vs Decision Problems**. Recall that a decision problem requires a YES/NO answer, and an optimization problem requires the "best possible answer", which often means maximizing or minimizing over some *cost* or *score*.

For most optimization problems, there is an obvious analogue as a decision problem. Namely, the decision problem takes an additional input k and outputs YES iff the input has an feasible solution of *score* at most k (for a minimization problem) or at least k (for a maximization problem) For example, consider the following problem:

MIN-VERTEX-COVER: Given a graph $G = (V, E)$, return the size of the smallest vertex cover in G.

MIN-VERTEX-COVER has a natural decision problem, namely VERTEX-COVER. In fact, every optimization problem can be converted to a decision problem in this way.

(a) Show that VERTEX-COVER $\leq_{\text{P}}$ MIN-VERTEX-COVER.

(b) Let B be an arbitrary optimization problem, and let A be the decision version of B. Show that

$$A \leq_{\text{P}} B\ .$$

(c) Show that MIN-VERTEX-COVER $\leq_{\text{P}}$ VERTEX-COVER.