CS41 Lab 12: NP-Completeness and approximation

November 23, 2021

This week, we'll continue exploring NP-COMPLETE decision problems, and develop approximation algorithms for related versions of those problems. You should focus on the first two problems.

1. **Travelling 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.

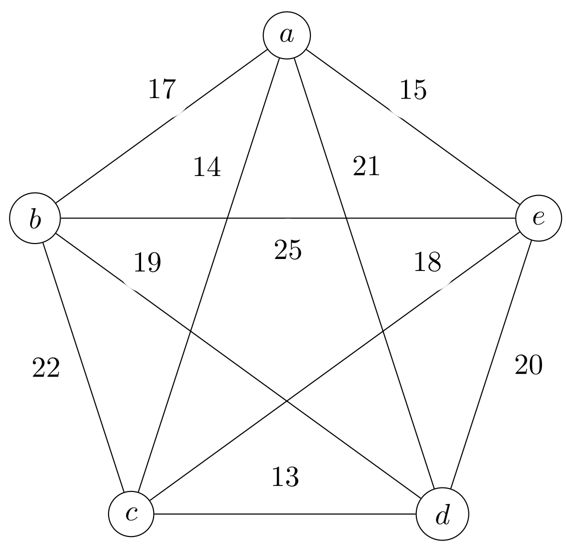
   For many TSP applications (such as when the cost is proportional to the distance between two cities), it makes sense for the edges to obey the *triangle inequality*: for every i, j, k, we have

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

   This version is often called METRIC-TSP.

   The (decision version of the) Travelling 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:



   (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: $\text{COST}(T) \leq \text{COST}(OPT)$.

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

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

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

2. Show that the following problems are $\in$ NP:

(a) TRAVELING-SALESMAN-DECISION. Given a complete graph $G = (V, E)$, nonnegative edge costs on all edges $\{c_e : e \in E\}$, and a target distance t, output YES iff it is possible for a travelling salesman to visit all the cities and return back home in a simple tour with total cost $\le t$.

(b) SUBSET-SUM. Given a list of n items with weights $w_1, \ldots, w_n$, along with a weight threshold W, output YES iff there exists a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i = W$.

(c) SET-COVER. The input is a set $\mathcal{U}$ of n elements, an integer k, and a collection $S_1, \ldots, S_m \subseteq \mathcal{U}$ of m subsets of $\mathcal{U}$. We want to output YES iff there is a set $T \subseteq \{1, \ldots, m\}$ with $|T| \le k$ such that

$$\bigcup_{i \in T} S_i = \mathcal{U}$$

(We think of the sets that T is picking as a "cover" of the universe of elements $\mathcal{U}$.)

3. **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.