CS41 Lab 11: approximations

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

1. **Toy storage.**

   Ika has lots of toys of all different sizes. You'd like to purchase a number of bins in which to store the toys. Approximately how many bins will you need?

   Let's formalize the toy storage problem as follows. Suppose there are n toys, with sizes $s_1, \ldots, s_n$, with $0 < s_i < 1$ for all i. Assume each bin has size 1 and can hold any collection of toys whose total size is less than or equal to 1.

   In this problem, you'll develop a greedy approximation algorithm, which works by taking each toy in turn and placing it into the first bin that can hold it. Let $S := \sum_{i=1}^{n} s_i$.

   (a) Show that the optimal number of bins is at least $\lceil S \rceil$.

   (b) Show that the greedy algorithm leaves at most one bin half full.

   (c) Prove that the number of bins used by the greedy algorithm is at most $\lceil 2S \rceil$.

   (d) Prove that the greedy algorithm is a 2-approximation algorithm for the toy storage problem.

2. **Travelling Salesperson Problem.** In this problem, a salesperson travels the country making sales pitches. The salesperson must visit n cities and then return to her home city, all while doing so as cheaply as possible, since travel between cities costs money.

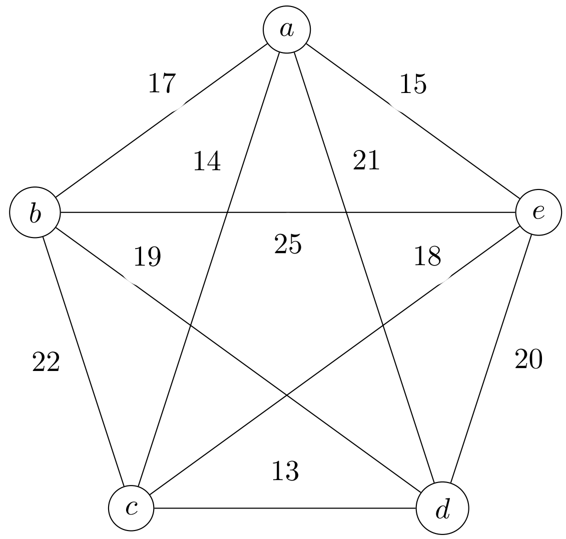
   The input is a complete graph $G = (V, E)$ along with nonnegative edge costs $\{c_e : e \in E\}$. A *tour* is a simple cycle $(v_{j_1}, \ldots, v_{j_n}, v_{j_1})$ that visits every vertex exactly once.[1] The goal is to output the minimum-cost tour.

   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
   $$c_{(ik)} \leqslant c_{(ij)} + c_{(jk)}.$$
   This version is often called METRIC-TSP.

   The (decision version of the) Travelling Salesperson 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: $\textsc{cost}(T) \leqslant \textsc{cost}(OPT)$.

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

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

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.