CS4234: Optimisation Algorithms Tutorial 2

MIN-SET-COVER, STEINER-TREE, M(W)FES

V1.2: Steven Halim *September 10, 2018*

Discussion Points

Q1: We discussed Integer Linear Programming (ILP) in (make-up) Lecture 02. Now express MIN-SET-COVER problem that we discussed in Lecture 03a as an ILP!

Q2: Now express MIN-SPANNING-TREE problem as an ILP! Will you solve MST problem that way?

Q3: We know that the distance function in the Euclidean plane is metric. Assume we have an algorithm for fully solving the METRIC-STEINER-TREE problem (okay, that is very hard, ... the approximate algorithm for METRIC-STEINER-TREE problem, like the one that we discussed in Lecture 03b). Can we (approximate) the EUCLIDEAN-STEINER-TREE Problem by reduction (we didn't do that in Lecture 03b, as I only showed reduction from GENERAL-STEINER-TREE to METRIC-STEINER-TREE (and back))? If so, how? If not, what is the problem?

Q4: Speaking of EUCLIDEAN-STEINER-TREE, let's spend some time discussing (and maybe proving) some of these properties that are discussed briefly in Lecture 03b. TA (Wei Liang) will draw a few (e.g. $n = 6$) 'random' points on Euclidean plane (the whiteboard) and we will try to apply these known properties to help us derive an optimal (or at least good enough) Euclidean Steiner Tree:

- Each Steiner point in an optimal solution has degree 3.
- The three lines entering a Steiner point form 120 degree angles, in an optimal solution.
- An optimal solution has at most $n - 2$ Steiner points.

Q5: Steven needs to continually increase the number of (NP-)hard problems that has been exposed to CS4234 students so far (to have more interesting Midterm Test and Final Assessment). So let's study this yet other problem MIN-WEIGHT-FEEDBACK-EDGE-SET (MWFES) and MIN-FEEDBACK-EDGE-SET (MFES) for the unweighted version.

You are given a *weighted* graph $G = (V, E)$ with weights $w : E \to \mathbb{R}$. Your goal is to delete the some edges to produce an acyclic graph with the maximum remaining edge weights. That is:

Find a minimum weight set of edges F such that $G = (V, E \setminus F)$ is acyclic.

Part 0. Draw an (small) example graph and explain this problem to your tutorial group.

Part 1. Assume G is an *undirected* graph. Give an efficient algorithm to solve this problem *optimally.*

Part 2. In the case of directed graphs, the MFES problem (both weighted or unweighted) is NP-hard (for now, just assume that it is really NP-hard and later in Part 4, we will show you the details). Now, consider the following (heuristic) algorithm: Given a directed graph $G = (V, E)$ and weights w where all edges have weight 1:

1. Let the vertices be $V = v_1, v_2, \ldots, v_n$.

2. Build graph $G_f = (V, E_f)$ containing only forward edges. That is, $E_f = (v_i, v_j)$ where $i < j$.
3. Build graph $G_b = (V, E_b)$ containing only backward edges. That is, $E_b = (v_j, v_i)$ where $i < j$.
4. Return the graph G_f or G_b containing higher weighted edges (or more edges in unweighted version) as the acyclic graph (and the other non selected edges as the removed edges F).

Show that:

- Both G_f and G_b are valid solutions to the MFES/MWFES problem.
- Show that the algorithm is *not* a good approximation algorithm for MFES/MWFES.

Part 3. Now consider the *complementary* (the dual) problem:

Find a maximum weight subgraph $G' \subset G$ that is acyclic.

Notice that *weight* here refers to the sum of the edge weights. Now show that the (heuristic) algorithm above is actually a 2-approximation algorithm.

Part 4 (optional, time permitting, otherwise, just read the modal answer). Assume G is a *directed* graph. Show that now it is NP-hard to solve by reduction from the VERTEX-COVER problem. That is, given an instance of VERTEX-COVER problem, show how to reduce it into an instance of FEEDBACK-EDGE-SET problem.