CS4234: Optimisation Algorithms	Tutorial 2

MSC, Steiner-Tree, M(W)FES, 2-CNF-SAT, PS2

V1.6: Steven Halim — *August 29, 2022*

MODAL ANSWER IS FOR OUR CLASS USAGE ONLY; NOT TO BE DISTRIBUTED IN PUBLIC

Discussion Points

Q1: We discussed Integer Linear Programming (ILP) in Lecture 02. Now express the proven NP-hard optimization problem: Min-Set-Cover problem that we discussed in Lecture 03a as an ILP!

Q2: Now express what we know as a P problem: Min-Spanning-Tree problem as an ILP! Will you solve MST problem that way instead of using what you already know from earlier modules?

Q3: In Lecture03b, we discussed the Euclidean-Steiner-Tree problem. Now let's spend some time discussing (and maybe proving) some of these properties that were only discussed briefly in Lecture 03b. The TA will draw a few (e.g., $n = 7$) '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 (visually) 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.

Q4: Steven needs to continually increase the number of (NP-)hard problems that has been exposed to CS4234 students so far (to have a 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 *directed 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. If G is an *undirected weighted* graph, give an efficient algorithm to solve this problem *optimally*.

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

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 more edges (or higher total weighted edges for the weighted MWFES 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 to save time, 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.

Q5: The `3-CNF-SAT` problem is one of the classic baseline problem to show the NP-hardness of a few other classic problems, e.g. `3-CNF-SAT` $\leq_p$ `Clique` $\leq_p$ `Vertex-Cover` as briefly shown in Lecture01. One of the PS2 'new' (2021) question involves a special case `2-CNF-SAT` of this `3-CNF-SAT`.

`2-CNF-SAT` problem is defined as follows: Given a conjunction of disjunctions ("and of ors") where each disjunction ("the or operation") has three (2) arguments that may be variables or the negation of variables, find a truth (T/F) assignment to these variables that makes the formula true. For example, given $\phi = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3)$ is satisfiable.

Part 1. Find a truth assignment to x_1, x_2 and x_3 so that the given ϕ is satisfiable.

Part 2. Which problem in PS2 is actually `2-CNF-SAT` problem?

Part 3. What is the high-level idea to solve this special case of an NP-complete decision problem?