CS4234: Optimisation Algorithms Tutorial 9

Stochastic-Local-Search, TSP, MWVC

V1.4: Steven Halim *October 26, 2020*

Preliminaries

During Lecture 9, we only discussed Travelling-Salesman-Problem (TSP) and a simple Hill Climbing local search algorithm for initial illustration of SLS ideas. Replace Hill Climbing (terminology for maximization problem) to Gradient/Steepest Descent for minimizing problem. In this tutorial, we will explore more about this TSP problem. Also, as your Mini Project involves another NP-hard optimization problem: Min-Weight-Vertex-Cover (MWVC), we will review one past paper question about this.

Discussion Points

Q1: For TSP, you have been exposed with 2-exchange (2-opt – notice that 2-opt here means 2-edge-exchange move, not 2-approximation) local neighborhood whereby we take 2 edges from the current tour, delete them, and reconnect the 2 tour fragments with the only other valid way, thus reintroducing 2 new edges. Some people (not necessarily from this module) actually misunderstand the meaning of 2-opt local move as a move that 'swaps 2 vertices/cities' of the current tour. Discuss what are the pros and cons of doing '2 edges swap' versus '2 vertices swap' in the context of TSP (and your Mini Project task 1), i.e., will you interpret 2-opt local move as 'swap 2 vertices'?

Q2: In lecture 9, Steven said that various TSP instances have different properties. So, please investigate Metric-No-Repeat TSP test case patterns. See `https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/` and perhaps try to visualize (or Google :O) the visualizations of some of these well known TSP benchmark test cases. What are the (initial) conclusions and/or local search ideas that you have after seeing all those?

Q3: Back in S1 AY 2017/18, Steven asked this in Final Assessment: Suppose that you are given several (unweighted simple) *general* graphs with n up to 1000 vertices and m ranging from 0 up to $n \times (n-1)/2$ (a complete graph). You are tasked to find the sizes of the Min-Vertex-Cover of those graphs (still the unweighted version). If the optimal answer for one of the graph G is x and your answer for that graph G is y, your score is 0 (for this entire question) if $y \geq 2x$. If your answer (y) is less than x (cannot be a valid vertex-cover as the optimal answer is x), you will also get 0. Otherwise, your score is $13.0 - (y - x)/x * 13.0$ (the context: This question worth 13 marks in the actual paper). Using *any* algorithm(s) or technique(s) that you have learned in this module, propose how you are going to deal with this scenario. You will be graded by the detail and soundness of your proposal. Assume that the time limit for your solution is 10s per test case.

Two AYs later, in S1 AY 2019/20, I finally have the digital form of this question: the `https://nus.kattis.com/problems/mwvc` — the Min Weight Vertex Cover problem as our Mini Project task 2. Ranald set n up to 4000, m up to 600 000, with time limit of 2s and similar auto partial scoring mechanism. So, let's discuss some initial ideas that you can use to get your task 2 up and running.