CS4234: Optimisation Algorithms Tutorial 8

Overview of Mini-Project, Stochastic-Local-Search, TSP

V1.6: Steven Halim *October 17, 2022*

Preliminaries

During Lecture 08 (on Week 09), we only discussed the Travelling-Salesman-Problem (TSP) and a simple Hill Climbing local search algorithm for the initial illustration of SLS ideas. We can replace Hill Climbing (the terminology for maximization problem) to Gradient/Steepest Descent for minimizing problem, or just call them Iterative-Improvement (II) for a more general term. In this tutorial, we will explore more about TSP.

However, because there is no tutorial on Monday of Week 10 (24 October 2022, due to Public Holiday: Deepavali), we have to discuss a bit more than usual in T08 to give a bit more guidance for the ongoing Mini Project. Thus, we will discuss a bit more hints about TSP (it is actually more suitable after listening to next week's lecture) and we will also discuss the other NP-hard optimization problem: Min-Weight-Vertex-Cover (MWVC) using one past paper question about this.

We will end this tutorial by reviewing the initial state of Mini Project `https://nus.kattis.com/courses/CS4234/CS4234_S1_AY2223/assignments/s23kxo`.

Discussion Points

Q1: Statements About SLS (up to Lecture 08 only) For each statement below about Stochastic Local Search (SLS) algorithm, determine if it is More Towards True/More Towards False/It depends and give a short explanation.

1. We can run an SLS algorithm (the first 'S' = Stochastic) for an NP-hard Combinatorial Optimization Problem (COP) instance for an **extremely long time**, e.g., $\approx \infty$, and still unable to prove that the best found solution of that run is the Global Optima (GO) for that COP instance.

2. SLS algorithms that use larger neighborhood is **always** better than SLS algorithms that use smaller neighborhood.

3. It may be possible to provide an approximation ratio for an SLS algorithm *that starts from any feasible solution* even when we only run the SLS algorithm for a finite amount of time. Current Computer Scientists are just not yet able to prove the approximation ratio of an SLS algorithm yet.

Q2: For TSP, you have been exposed with 2-exchange (or 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'?

Q3: [Speed up ahead] Later in Lecture 10, Steven will say that various (still NP-hard) TSP instances have different properties that can be 'attacked differently'. So, please investigate Metric-No-Repeat TSP test case patterns. See `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/` 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?

Q4: 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.