Northeastern University
CS7180 – Algorithms
Spring 2024 Semester, Jonathan Mwaura

Problem Set #2

This Problem Set #2 is due at **11.30PM on 19th, Feb 2024** , and will be submitted via Canvas.

This Problem Set will be marked out of 100, and is worth 8% of the final course grade.

This problem set is to be completed **individually** unless otherwise directed.

Please type your solutions on standard 8.5×11 paper, with your name at the top of each solution. Ensure that you submit you solutions in one file PDF file on Gradescope. **each problem sets solution should be on in its own individual page, Gradescope will help ensure you submit each solution under its correct problem number**

Make sure you label your Problem Set #2 submissions appropriately using your lastname-1 (i.e. lastname 'hypen' number of problem set number) - e.g. Mwaura-1a.pdf.

While a solution must be absolutely perfect to receive full marks, I will be generous in awarding partial marks for incomplete solutions that demonstrate progress.

So that there is no ambiguity, there are two non-negotiable rules. A violation of either rule constitutes plagiarism and will result in you receiving an F for this course.

(a) If you meet with a classmate to discuss any of the Individual Problems, your submission must be an individual activity, done in your own words, away from others. The process of finding a solution might take 3 - 5 iterations or even more BUT you learn from all these attempts and your confidence grows with each iteration.

(b) These problem sets might seem hard on a first look. They are designed to be so. We learn by attempting problems, struggling through them and coming on top. I encourage you to make this learning exercise worth your while. What do I mean? Open the problem sets as early as you get them, then do not look at hints or answers any where (including on the internet and consulting other students for direct answers), give it the best shot you can. If you get stuck come to Professor or TA's office hour and we shall be glad to listen to your rationale and work with you till you are able to tackle the problem sets.

Problem #1 – INDIVIDUAL

In problem set 1, you explored the concept of NP completeness and completed some exercises using the Travelling Salesman Probem (TSP) as a problem domain. In this problem set, we shall explore both TSP and 0/1 knapsack problems using newer techniques that we have learnt in class. These include hill climbing techniques and simulated annealing as well as evolutionary algorithms (including genetic algorithms and different EA variants). For these two problems we shall use dynamic programming solutions as the 'ground truth'

(a) The assumption here is that everyone has now completed the Held-Karp DP algorithm for the TSP problem. In this part of the question, make sure the algorithm runs on the NEU's Discovery cluster and that you are able to generate results for up to 100 cities.

(b) The knapsack problem (also called backpack problem) is a well known combinatorial optimization problem. See this wiki : https://en.wikipedia.org/wiki/Knapsack_problem for a discussion about the problem. There is a pseudo-polynomial time algorithm for solving this problem using Dynamic Programming. Please see Abdul Bari's video https://www.youtube.com/watch?v=nLmhmB6NzcM to see how this is done (You can also consult Prof Mwaura regarding the technique). In this part of the question, you will need to implement the DP algorithm to solve the P01 knapsack problem found here: https://people.sc.fsu.edu/ jburkardt/datasets/knapsack_01/knapsack_01.html. Please report the solution you found.

(c) Below are seven metaheuristics that we are discussing in class. Your work is to select one of this techniques and implement it for both the TSP and Knapsack problems. Note that, your algorithm selection must be unique to your work. **I shall allow for collaborative efforts for related techniques**, for instance the hill climbing and simulated annealing (as a group) and the evolutionary algorithms (as another group). However, once common development is done, each person must evaluate and tune implementation to their specific technique.

(i) Hill Climbing (Steepest Ascent and First Choice, (ii) Stochastic, Random Restart)
(iii) Simulated Annealing
(iv) Generational Genetic Algorithms
(v) Steady State Genetic Algorithms
(vi) Differential Evolution
(vii) Evolutionary Strategies

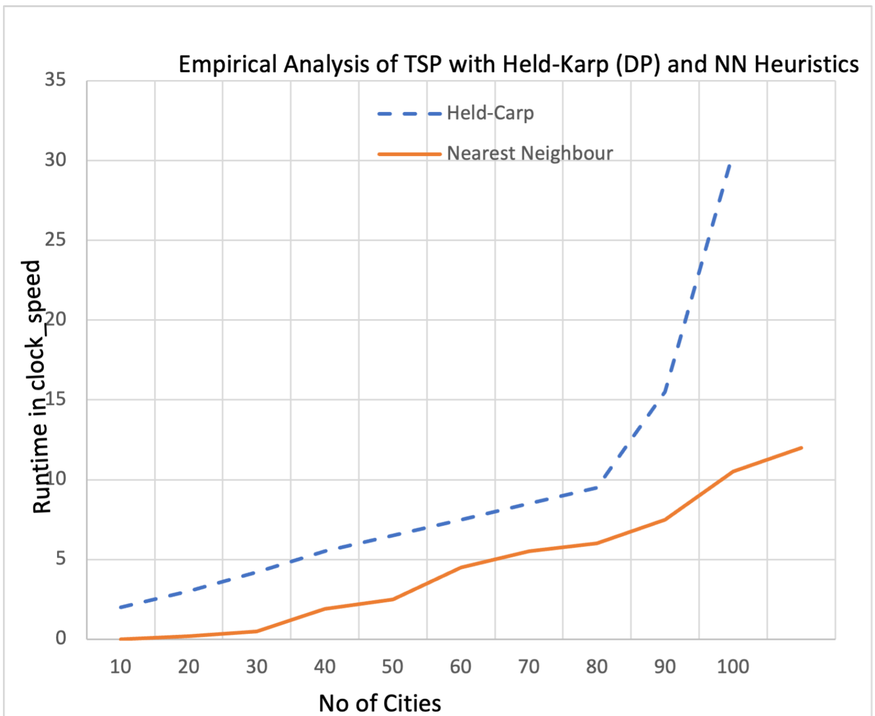
(d) Empirical Analysis (TSP): You will now need to run both your DP from (a) above and your metaheuristic from (b) for the TSP. Similar to Problem Set 1, we would like to see the run time based on how many cities are in the TSP. You will need to go to this site: https://github.com/acu192/fun-tsp-challenge. There are 4 different TSP datasets, for our question we are interested in the medium data set (https://github.com/acu192/fun-tsp-challenge). This contains a TSP with 100 cities. However, we want to work with it progressively. So we shall start with 10 cities and increment this with 10 cities until we reach 100 cities.

For each run we need to record the time spent (note in order to remove any randomness, we would like to repeat each experiment for 20 runs and record the average). Once we have the data (i.e. 10 rows - each problem per row and the average run time) we shall need to plot on Average

runtime vs no of cities (see Problem below).
You are expected to output a graphical display of the running times as shown in the figure below. Additionally, you will need to report and discuss your intuition into these running times versus the actual fitness that you obtained, i.e. a direct comparison between your chosen technique and the DP technique. For instance:

(i) does your algorithm perform better with smaller city sizes? what about the larger ones?

(ii) DP will give an exact result, if you were to take the average solution achieved with your algorithm (averaged from the 20 runs per each problem), what is the do you obtain Relative approximation or Absolute approximation (see notes from class - slides on Approximation from Unit 1).



(e) Empirical Analysis (Knapsack Problem): In this part of the question, we are intersted in checking how well our technique works and how close its solution is compared to the DP solution. You will need to run your DP and your metaheuristic across P01 to P08. For each of these, repeat the experimentation for 20 independent runs to ensure that the results are not affected by randomness. Since the DP is an exact solution, it is to be expected that DP's results will remain consistent throughout your runs. However, the meta-heuristics results will vary. For each of the P01 - P08 report the average accuracy for solving the problem using your chosen heuristic. Does the heuristic do better in smaller or larger problems?

(f) Reflection and discussion: In this part reflection on your work and discuss what you think could be adapted in your technique in order to generate better solution. Also check with the literature to see what other authors/researchers have said about this technique. Do your results conquer with them?