MA4254/AY21-22 – Discrete Optimization
Computational Assignment

September 17, 2021

Instructions

1. This project assignment accounts for 20% of your total grade.
2. It is due 29th October 2021, 4pm (Friday, Week 11).
3. You are to submit a report and your code. In addition to responses to all questions in this assignment, your report should also include a write-up that
   (a) Explain the steps you did when performing this computational project
   (b) Explain implementation details
   (c) Your results as well as your interpretations and conclusions from these results

   Graphs and figures are an excellent way to present your solutions. In all, an appropriate length of your report is about 3 pages per question.
4. Submission rules: Include all files into a single ZIP file. You should name your file with your matriculation number. Submit the file onto the LumiNUS submission folder.
5. You are required to include your code in your submission. The code does not count into the page length of your report. You are permitted to code in any language you wish.
6. Please strive to write your report as neatly as possible. LATEX is the recommended (and simplest) option, but not strictly required. If you need help, please write to me or consult your classmates.
7. You are allowed to *verbally* discuss the assignment with your coursemates. You are *not* allowed to read each other's written answers.
8. You *must* write your own code. You can, however, discuss coding techniques or difficulties with your classmates.

9. Cite all references clearly. This include Internet references.

10. You are not allowed to consult external solutions; e.g., you *cannot* consult a piece of code that directly solves a question in the assignment. If in doubt, check with me.

---

1. Facility Location Problem

Recall the facility location problem: Suppose that there are n facilities and m customers who require a service from one of these facilities. For a facility to be able to provide the service, the facility needs to be "switched on", for a cost of c_j. The customers selects, among all the facility that are "switched on", to be serviced by a single facility at a cost of $d_{i,j}$. The objective is to minimize the total cost comprising switching on facilities and travel costs, subject to the constraint that every customer is serviced.

This problem can be modeled as a MILP. In the following, the decision variables are y_j, which models whether facility j is "switched on", and $x_{i,j}$, which models whether customer i selects facility j for servicing.

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} c_j y_j + \sum_{i,j=1}^{m,n} d_{i,j} x_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{i,j} = 1 \qquad \text{for all } i \\ & x_{i,j} \le y_j \qquad \text{for all } i, j \\ & 0 \le x_{i,j} \le 1, y_j \in \{0,1\}. \end{aligned} \tag{FLP}$$

An alternative and equivalent MILP formulation is to replace the constraint $x_{i,j} \le y_j$ with the constraint $\sum_i x_{i,j} \le m y_j$.

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} c_j y_j + \sum_{i,j=1}^{m,n} d_{i,j} x_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{i,j} = 1 \qquad \text{for all } i \\ & \sum_{i=1}^{m} x_{i,j} \le m y_j \qquad \text{for all } j \\ & 0 \le x_{i,j} \le 1, y_j \in \{0,1\}. \end{aligned} \tag{AFL}$$

The latter formulation (AFL) is known as the aggregate facility location problem.

(a) Count the number of linear inequalities in (FLP) and (AFL). Which formulation has fewer number of inequalities?

(b) Show that the set of feasible solutions to (FLP) and (AFL) are equal. Conclude that these two formulations are equivalent.

(c) Derive the linear relaxation to (FLP) by replacing the binary constraints $y_j \in \{0, 1\}$ with the box constraint $0 \leq y_j \leq 1$. Denote the linear relaxations by (FLP-LR).

(d) Derive the linear relaxation to (AFL) by replacing the binary constraints $y_j \in \{0, 1\}$ with the box constraint $0 \leq y_j \leq 1$. Denote the linear relaxations by (AFL-LR).

(e) Denote the optimal value to (FLP) and (AFL) by (FLP-Val) and (AFL-Val) respectively. Denote the optimal value to (FLP-LR) and (AFL-LR) by (FLP-LR-Val) and (AFL-LR-Val) respectively. What is the relationship between (FLP-Val), (AFL-Val), (FLP-LR-Val), and (AFL-LR-Val)? Specifically, which one is bigger, in general? Justify your answer.

(f) Next, we generate random instances of the facility location problem. For this, it is recommended that you pick $n = 15$ and $m = 20$ (m should be larger than n). Pick n locations uniformly at random from the square $[0, 1] \times [0, 1]$ to model the locations of the facilities. Draw n random variables from the $\mathsf{Unif}[0, 1]$ distribution to model the cost. Draw an additional m locations uniformly at random from the square $[0, 1] \times [0, 1]$ to model the locations of the customers. The distance $d_{i,j}$ is the Euclidean distance between the i-th customer and the j-th facility. Calculate the distance matrix between every customer and facility.

(g) Using the generated dataset, solve the facility location problem using (FLP) and (AFL). Solve the linear relaxations (FLP-LR) and (AFL-LR) corresponding to these formulations. Record the optimal value to all four instances. Repeat this process 100 times with different realizations of all random variables (in other words, repeat steps (f) and (g) 100 times).

(h) Compare and comment on the difference between the optimal values of all four optimization instances. How often is (FLP-Val) equal to (FLP-LR-Val)? How often is (AFL-Val) equal to (AFL-LR-Val)?

2. Traveling Salesman Problem

The Traveling Salesman Problem seeks the shortest possible tour while visiting every location exactly once, and returning to the starting location.

Concretely, suppose that there are n locations indexed by $\{1, 2, \ldots, n\}$. Let d_{ij} be the matrix whose (i, j)-th entry denotes the distance between location i and j. A tour is represented by a *permutation* $\{a_1, a_2, \ldots, a_n\}$ of the sequence $\{1, 2, \ldots, n\}$. As such, a tour visits all locations in the following order:

$$a_1 \to a_2 \to \ldots a_{n-1} \to a_n \to a_1.$$

In this problem, we will be implementing the Simulated Annealing algorithm for solving the Traveling Salesman Problem.

(a) Create a function that takes as input a tour and outputs the total distance

$$f(\{a_1, a_2, \ldots, a_n\}) = d_{a_1,a_2} + d_{a_2,a_3} + \ldots + d_{a_{n-1},a_n} + d_{a_n,a_1}.$$

Remember to return to the starting location!

(b) We need to create a function that takes as input a tour, and outputs a perturbed tour. These perturbed tours are candidate tours that the algorithm will evaluate and decide to accept or reject. We generate a candidate tour as follows. Suppose the input tour is the following sequence:

$$\{a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{j-1}, a_j, a_{j+1}, \ldots, a_n\}$$

Pick indices $1 \leq i < j \leq n$ uniformly at random. The candidate tour is the following:

$$\{a_1, a_2, \ldots, a_{i-1}, a_j, a_{j-1}, \ldots, a_{i+1}, a_i, a_{j+1}, \ldots, a_n\}.$$

(c) Implement the following: Initialize with a random tour. Initialize with some choice of temperature parameter $T > 0$. Initialize with some choice of temperature update parameter $0 < \eta < 1$. At *every* iteration, you should perform the following sequence of steps:

1. Let f_{curr} denote the length of the current tour.

2. Propose a randomly chosen new candidate tour. Let f_{cand} denote the length of the new tour.

3. Draw a random variable $u \sim \mathsf{Unif}[0, 1]$. If the following condition holds

$$\exp\left(-\frac{f_{\text{cand}} - f_{\text{curr}}}{T}\right) \geq u,$$

replace the current tour with the candidate tour. If the condition does not hold, simply proceed.

4. Update the choice of temperature with

$$T \leftarrow \eta T.$$

An instance of the simulated annealing algorithm for solving the TSP problem performs multiple loops of the above sequence of steps.

(d) Generate a random instance to test your implementation. Pick $n = 100$. Generate n cities uniformly at random over the $[0, 1] \times [0, 1]$ grid. Let the pairwise distances between cities be the Euclidean distance.

Implement the Simulated Annealing algorithm described above to find the shortest tour. Do note that the Simulated Annealing algorithm is a *heuristic* – while it is very effective at finding high quality solutions, it is not able to certify optimality of any solution. So, generally speaking, you will *not* be able to tell if your solution is globally optimal.

Suggestions: Try $\eta = 0.99$. Run for about 10000 loops. Experiment with a few choices of initial temperature T.

(e) Plot the cities with the tour to show your results. An optimal tour does not edges that intersect. Does your tour have such edges?

(f) Experiment with a different proposal rule in which we pick a pair of indices $i < j$, and we swap the indices i and j while leaving all other indices (including those between i and j) intact. How does this choice of proposal rule compare with the previous one?

(g) *Fun challenge.* Download the file *busstops.csv* from LumiNUS. Use these locations as the cities. Apply your simulated annealing algorithm and report the shortest tour you are able to find (*xloc* and *yloc*). Submit the shortest sequence you were able to find (see *submitTSPbusstop.csv* as a guide). Post your distance (distance only!) on the LumiNUS forum as a challenge to your classmates. The winners will announced after the project is due.

Submission guideline. You are to submit a csv file with the sequence of id's that your tour visits. Do *not* use an internal naming system. I will use your files to verify the distance of your tour. Check with the posted file *submitTSPbusstop.csv* as a guide.

Note. There are 5000 plus data points in this set, which makes this task substantially more challenging than the previous problem. You are *only* required to submit a sequence for this problem. There is absolutely no other requirement for the quality of your solution – you will get full credit so long as you satisfy the submission guidelines. Only fame and reputation is at stake here.

Thanks. The dataset is the location of all Bus Stops in Singapore, and is obtained from LTA DataMall:

`https://datamall.lta.gov.sg/content/datamall/en/static-data.html`