Tabu Search

Due Date: 11:59pm, July 5, 2024

1 Problem (20 points)

This is one of the QAP (quadratic assignment problem) test problems of Nugent et al. 20 departments are to be placed in 20 locations with five in each row (see below). The objective is to minimize costs between the placed departments. The cost is (flow * rectilinear distance), where both flow and distance are symmetric between any given pair of departments. The flow and distance matrices are on Learn. The optimal solution is 1285.

(note: rectilinear is also known as Manhattan distance)

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

1. Code a simple TS to solve the problem. To do this you need to encode the problem as a permutation, define a neighborhood and a move operator, set a tabu list size and select a stopping criterion. Use only a recency based tabu list and no aspiration criteria at this point. Set the default neighborhood function to check the whole neighborhood. (6 points.)
2. Run your TS. (2 points)
3. Perform the following changes on your TS code (one by one) and compare the results. When moving on to the next task, you don't need to repeat the experiment from previous tasks.
    - Change the initial starting point (initial solution) 10 times, the solutions should be randomly generated. (2 points)
    - Change the tabu tenure – once smaller and once larger than your original choice. (2 poitns)
    - Change the tabu tenure to a dynamic one – an easy way to do this is to choose a range and generate a random uniform integer between this range every so often (i.e., only change the tabu list size infrequently). (2 points)

- Add 2 aspiration criteria in 2 separate experiments: best solution so far, best solution in the neighborhood. (2 points)
- Use a subset of the whole neighborhood to select the next solution. (2 points)
- Add a frequency based tabu list to encourage the search to diversify. That is, record the number of times each move has been performed. Then each time, encourage moves that have been selected a few times. (2 points)

Submission Requirements:

- Code: Provide your implementations for the required functions, step by step using Jupyter Notebook. Upload your .ipynb file to the Learn dropbox.
- Report: Submit a detailed report containing necessary explanations, analyses, experimental results, graphs, and your findings as a PDF to the Learn dropbox.

Notes for Students:

- Ensure your code is well-documented and includes comments explaining key parts of your implementation.
- Your report should be well-structured, with clear sections for each part of the assignment.
- In your presentation, focus on clarity and conciseness, highlighting the most important aspects of your work and the points being asked by the question.