CENG 319

ALGORITHM ANALYSIS

Fall 2023-2024

Project Delivery:

January 11, 2024 at 23:59

Notes

- The project will be done individually or in a group that has at most 3 students.
- The project report should be created in ".pdf" format.
- Important code structures used in coding can be shared in the project report. *** Do not share all codes in the report.
- Share all coding via Github to
- Create a video recording of a short simulation and introduction of the project, a maximum of 10 minutes.
- Submit the project report and video recording as a single archive file (zip) using the project submission form.

Project Details:

Input: US backbone topology which consists of 24 nodes and 43 edges is given as input. The input file contains adjacency, bandwidth, delay, and reliability matrices. The relationship between each node in these matrices is separated by “:”. Each matrix is separated by an empty row.

- In the neighborhood matrix, link distances are uniformly distributed between 1 and 5.
- The bandwidth matrix uniform distribution is between 3 and 10.
- The delay matrix is uniformly distributed between 1 and 5.
- The reliability matrix is between 0.95 and 0.99.

Request: A request will have source node id, destination node id, and bandwidth requirement information.

Solution: The function, called Solution, will take the input txt file and the defined request. With the information taken as parameters, it is desired to find the shortest path between two points (source and destination) by satisfying the following constraints.

You have to do the requirement. When you add delay and reliability constraints to the solution, you can get additional points.

- **Constraints:** The solution must be within the range of the specified constraints.
    - Bandwidth demand must be 5. **(required)**
    - Delay threshold must be less than 40. **(extra 5 points)**
    - Reliability threshold must be greater than 0.70. **(extra 5 points)**

Algorithm: Three of the following algorithms must be used to solve the problem.

- Dijkstra Algorithm
- Bellman-Ford Algorithm
- A* Algorithm
- Flody-Warshall Algorithm

Additional points will be awarded for solving the problem with any of the meta-heuristic algorithms specified. **(extra 10 points)**

- Simulated Annealing Algorithm
- Tabu Search Algorithm
- Ant Colony Algorithm
- Bee Colony Algorithm
- Firefly Algorithm

Explanation:

- First, you should read the input file and take the parts suitable for the constraint you will use.
- Then, you need to send the parameters to the method according to your request and find the path that provides the necessary constraints. If you want to find a solution for more than one constraint, you must make sure that the path found satisfies all constraints.
- Each method should be written in separate methods. If you are making a solution with multiple parameters, you need to individually specify methods for each one.

Output: After running the algorithm, it should give all nodes in the calculated path.
The problem should find a solution according to the objective function given below.

$$min\left(bw \times \sum_{ij\,\in E} \delta_{ij} \times dist_{ij}\right)$$

In the given equation:

- Graph (G) consists of vertex and edges, G=(V,E). i and j are two vertices defined in the graph and ij denotes an edge in the edge set.

  bw represents bandwidth demand,

  $$\delta_{ij} = \begin{cases} 1, if\ edge\ (i,j) is\ used \\ 0, \qquad\qquad otherwise \end{cases}$$

  $dist_{ij}$, edge distances between i and j.