Extended Task Report: Social Network Analysis

Bharat Sharma
Roll No: 240265

June 4, 2025

1 Objective

This task aimed to simulate a small-scale social network of students and explore its properties using graph-theoretic tools.The graph was generated using the following rules:

- Students are divided into 5 equal sections.
- 25 popular students are selected at random.
- Edges are added within sections with a probability of 0.08.
- A small number (20 to 50) of cross-section edges are sampled and connected with a probability based on popularity.
- All edge weights are integers between 1 and 10.

2 Graph Generation

To build the network, I divided students evenly into sections A through E. Each student was represented as a node in an undirected graph. Friendships were modeled as weighted edges, with 25 students randomly marked as "popular," giving them a higher chance of connecting with others.

Edges within the same section were added probabilistically (with a base chance of 0.08), and the popularity of students increased these odds. A small number of inter-section friendships (ranging from 20 to 50 randomly chosen pairs) were introduced, influenced by popularity. Edge weights, indicating friendship strength, were randomly chosen from integers between 1 and 10.

3 Friend Group Patterns

Once the graph was constructed, I analyzed the graph structure.

In some cases, entire sections formed dense groups, while in others, several small clusters emerged. Occasionally, students were isolated entirely, without any connections. Popular students frequently served as links between otherwise disconnected parts of the graph, showing their influence in increasing connectivity. No two runs produced the same result, making each simulation unique and highlighting how randomness and popularity influence social cohesion.

4 Shortest Path Algorithms

To study how easily students could reach each other, I selected five random student pairs and computed shortest paths using Dijkstra's algorithm and the A* algorithm.

Dijkstra's method found the optimal path by considering all possibilities, while A* used a basic heuristic: assigning zero cost if students were in the same section and a cost of five otherwise. This heuristic helped A* prioritize relevant paths without significantly impacting accuracy.

Both methods produced the same results in terms of path and weight in most cases. However, A* was able to reach the solution slightly faster when the heuristic provided helpful guidance, especially for students in different sections. This emphasized how structured assumptions about the network (like section grouping) can enhance algorithmic performance.

5 Conclusion

By incorporating ideas like student popularity and section-wise grouping, the network reflected familiar social patterns, including tight-knit circles, connectors, and students on the periphery. Running shortest path algorithms provided further perspective on how information or interaction might travel through such a network. Even a simple heuristic in the A* algorithm made a difference in navigating the structure more efficiently.