# Graph Centrality Analysis and Genetic Algorithm Optimization

## 1. Introduction

Graph analysis plays a fundamental role in understanding the influence of nodes in a network. Various centrality measures help quantify the importance of nodes based on structural properties. However, selecting the most influential nodes using predefined weights may not always yield optimal results. This study employs a **genetic algorithm (GA)** to optimize centrality weights, improving the selection of key nodes in a network.

The optimization process aims to maximize the variance of weighted centrality scores, enhancing the differentiation between influential nodes. Such an approach is particularly valuable in applications like **social network analysis**, **transportation planning**, and **biological network studies**, where identifying key nodes is critical.

## 2. Graph Generation

To create a suitable network for analysis, a **connected, undirected, and simple graph** is generated with the following characteristics:

- **Node Count**: 1000 nodes.
- **Spanning Tree Formation**: A minimum spanning tree ensures basic connectivity among nodes.
- **Random Edge Addition**: Additional edges are introduced probabilistically to resemble real-world network structures.

This approach ensures a realistic network topology while maintaining computational efficiency for centrality calculations and GA-based optimization.

## 3. Centrality Measures

Four key centrality measures are considered, each capturing different aspects of node influence:

1. **Degree Centrality ($C_D$)**:
   a. Measures the number of direct connections a node has.
   b. High-degree nodes are often considered influential in spreading information.
2. **Closeness Centrality ($C_C$)**:
   a. Evaluates how efficiently a node can reach all others in the network.
   b. Nodes with high closeness centrality are well-positioned for fast communication.
3. **Betweenness Centrality ($C_B$)**:
   a. Captures nodes that frequently appear in the shortest paths between other nodes.
   b. Such nodes act as "bridges" and can control the flow of information.
4. **Eigenvector Centrality ($C_E$)**:
   a. Considers not just a node's connections but also the importance of its neighbours.
   b. Nodes with high eigenvector centrality exert influence over other influential nodes.

Each centrality measure is normalized to the range **[0,1]** for consistent comparison and weight optimization.

## 4. Optimization Using Genetic Algorithm

A **genetic algorithm (GA)** is implemented to optimize the weights of the four centrality measures, ensuring that the ranking of nodes better reflects their actual importance in the network.

### GA Setup:

- **Objective**: Maximize the standard deviation of weighted centrality scores to improve node differentiation.
- **Chromosome Representation**: A set of four weights $(w_1, w_2, w_3, w_4)$ corresponding to $C_D$, $C_C$, $C_B$ and $C_E$ respectively.
- **Fitness Function**:

  - Computes the weighted sum of centrality scores for all nodes.
  - Evaluates the standard deviation of scores across nodes.
- **GA Parameters**:
  - **Population Size**: 20 individuals.
  - **Generations**: 50.
  - **Selection**: Tournament selection (selects better individuals with high probability).
  - **Crossover**: Blend crossover ($\alpha = 0.5$) for smooth weight adjustment.
  - **Mutation**: Gaussian mutation to introduce variability.

The GA iteratively evolves better weight combinations, leading to improved differentiation in ranking.

# 5. Results & Observations

## 5.1 Top 10 Nodes Without Optimization

- Initially, nodes are ranked using equal weights $(1,1,1,1)$.
- The ranking provides a broad measure of influence but lacks fine differentiation.

## 5.2 Top 10 Nodes After GA Optimization

- The GA finds an **optimal weight combination** that enhances node ranking.
- Key observations:
  - The **gap between the 10th and 11th ranked nodes increases**, making ranking clearer.
  - Some nodes that were previously underestimated rise in rank, revealing hidden influential nodes.
  - The network's structure becomes more evident, highlighting influential clusters.

These results suggest that **optimizing centrality weights significantly improves the identification of key nodes** in the network.

# 6. Graph Visualization

Three graphical representations are generated to illustrate the process:

1. **Original Graph**: Displays the overall network structure.
2. **Top 10 Nodes (Equal Weights)**: Highlights the most influential nodes before GA optimization.
3. **Top 10 Nodes (Optimized Weights)**: Shows refined rankings with optimized weight distribution.

The comparison between visualizations highlights the improvement in node identification, emphasizing the effectiveness of GA-based optimization.

## Output:

- **<u>Output 1:</u>**

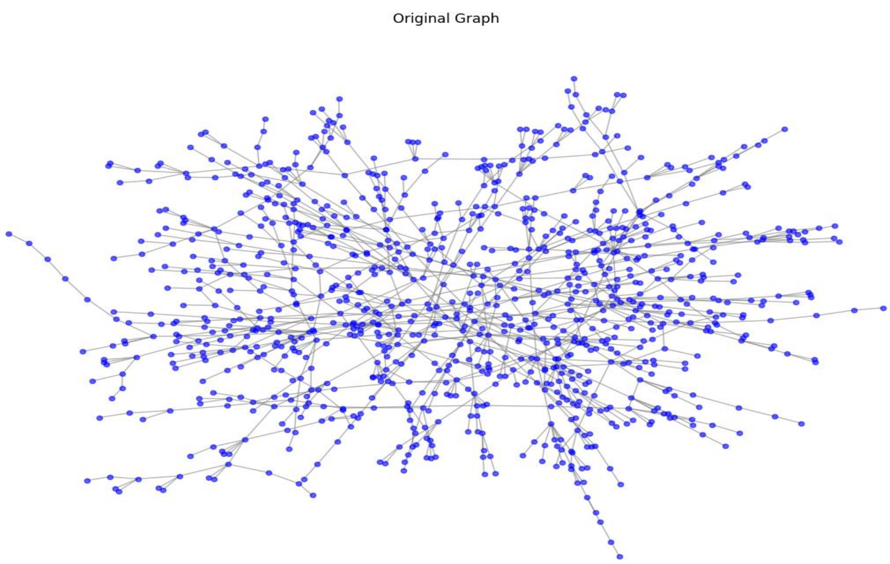


**Fig 1.1: Graph BEFORE detecting top 10 nodes**


**Fig 1.2: Graph AFTER detecting top 10 nodes**

## RESULT 1:

```
Optimal Weights (Degree, Closeness, Betweenness, Eigenvector): [1.9829510484106843, 2.0493001666880617, 1.2558292124646633, 1.5081195512423278]
```

```
Top 10 Nodes Without Optimization:
Node: 2, Score: 4.0
Node: 3, Score: 2.4628842786515324
Node: 0, Score: 2.3911617441786133
Node: 1, Score: 2.3742972178274613
Node: 7, Score: 2.253574179395879
Node: 9, Score: 2.030251630286944
Node: 5, Score: 1.965212049249247
Node: 24, Score: 1.9641596934477492
Node: 6, Score: 1.9559224345347852
Node: 11, Score: 1.868941822820394
```

```
Top 10 Nodes With Optimization:
Node: 2, Score: 6.7961999788057375
Node: 0, Score: 4.313436742799054
Node: 3, Score: 4.283472178411029
Node: 7, Score: 4.116228476831822
Node: 1, Score: 4.050242511982354
Node: 9, Score: 3.735797978461175
Node: 24, Score: 3.584357887477661
Node: 6, Score: 3.554121996781141
Node: 5, Score: 3.521568422066083
Node: 11, Score: 3.446368673420582
```

- **<u>Output 2:</u>**

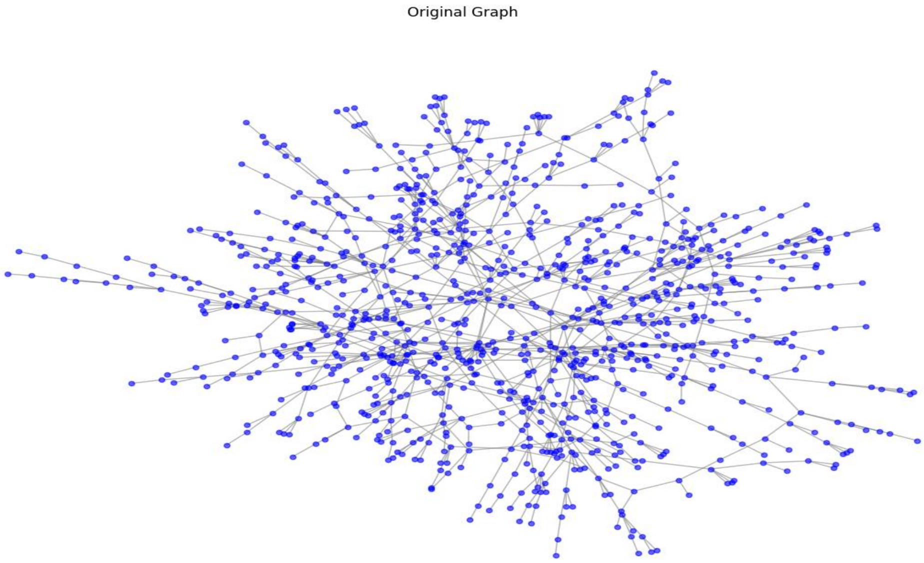


**Fig 2.1: Graph BEFORE detecting top 10 nodes**


**Fig 1.1: Graph AFTER detecting top 10 nodes**

**RESULT 2:**

```
Optimal Weights (Degree, Closeness, Betweenness, Eigenvector): [2.288037347891194, 1.7522632409605066, 2.7490245523598937, 1.9460325933087126]
```

```
Top 10 Nodes Without Optimization:
Node: 1, Score: 3.8
Node: 0, Score: 3.4431746202061118
Node: 5, Score: 2.808421093474931
Node: 3, Score: 2.5773132911737924
Node: 2, Score: 2.471228927237215
Node: 4, Score: 2.1924092738143814
Node: 24, Score: 1.9432197000982139
Node: 6, Score: 1.8344454370246788
Node: 96, Score: 1.8097718070716933
Node: 49, Score: 1.7800504682939453
```

```
Top 10 Nodes With Optimization:
Node: 1, Score: 8.27775026494207
Node: 0, Score: 7.369549867755266
Node: 5, Score: 5.945325554104446
Node: 3, Score: 5.494213857049024
Node: 2, Score: 5.2318340068288025
Node: 4, Score: 4.710355072216152
Node: 24, Score: 4.017159113789612
Node: 6, Score: 3.784907534712684
Node: 96, Score: 3.7230612968738392
Node: 37, Score: 3.5745873288668406
```

# 7. Conclusion

This study demonstrates that **genetic algorithms can effectively refine node ranking in network analysis**. By optimizing centrality weights, the GA enhances differentiation between key nodes, leading to more meaningful insights into node importance.

## Key Takeaways:

- The **standard method (equal weights)** provides a general ranking but lacks precision.
- **GA-based optimization** finds superior weight combinations, improving ranking clarity.
- The optimized rankings can be applied to **network resilience analysis**, **social influence detection**, and **transportation systems**.

Future work may explore:

- **Multi-objective optimization**: Balancing multiple network properties.
- **Dynamic graphs**: Adapting weights in evolving networks.
- **Hybrid AI techniques**: Combining GAs with machine learning for deeper insights.

This research highlights the power of optimization in graph analysis, offering a **robust methodology for identifying influential nodes** in complex networks.