# DATA STRUCTURES AND ALGORITHMS IDEATHON REPORT

**PRAGATI SINHA (B22MT048)**
**SHUBHAM KUMAR (B22CI038)**
**PALAK BHAWSAR (B22CH018)**

## *COMMUNITY DETECTION USING DIFFERENT ALGORITHMS*

**INTRODUCTION:**

Community detection is a fundamental task in network analysis, crucial for understanding the structure and dynamics of complex systems. The goal of community detection algorithms is to identify cohesive groups of nodes within a network, known as communities or clusters. In this report, we explore and compare the performance of three popular community detection algorithms: Girvan-Newman, Label Propagation, and Modularity algorithms.

**Problem Statement:**

The problem statement revolves around the need to effectively detect communities within complex networks, such as social networks, biological networks, and technological networks. Understanding the performance of various community detection algorithms and their scalability is crucial for applications ranging from social network analysis to recommendation systems and community detection in biological networks.

Dataset used: Bitcoin Alpha Trust Weighted Signed Network
Python Libraries used:

```
import networkx as nx
import matplotlib.pyplot as plt
from networkx.algorithms import community
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
from sklearn.metrics.cluster import normalized_mutual_info_score
import random
import csv
import time as t
```

Loading the dataset:

```
path="/content/drive/MyDrive/soc-sign-bitcoinalpha.csv"
```

Creating a graph:

```
G_og = nx.Graph()
with open(path, 'r') as f:
    reader = csv.reader(f)
    for row in reader:
        G_og.add_edge(row[0], row[1])
all_edges = list(G_og.edges())
```

This graph has 24,168 edges and 3783 nodes. We take 7000 edges at first and iteratively decrease the edge count by 1000. We perform community detection for each iteration. We have then plotted the communities detected via the three algorithms alongside the original graph for each iteration.

**Methodology:**

We have used 3 algorithms for community detection in this code-

1. The Girvan-Newman algorithm

```
def girvan_newman(G):
    communities = community.girvan_newman(G)
    return tuple(sorted(c) for c in next(communities))
```

The Girvan-Newman algorithm is a hierarchical method for detecting communities in complex networks by iteratively removing edges with the highest betweenness centrality.

2. The Label Propagation algorithm

```
def label_propagation(G):
    communities = community.asyn_lpa_communities(G)
    return tuple(sorted(c) for c in communities)
```

Label Propagation algorithm assigns labels to nodes in a network based on the majority label of its neighbors, often used for community detection in large-scale graphs.

3. Modularity Community Detection algorithm

```
def modularity(G):
    communities = community.greedy_modularity_communities(G)
    return tuple(sorted(c) for c in communities)
```

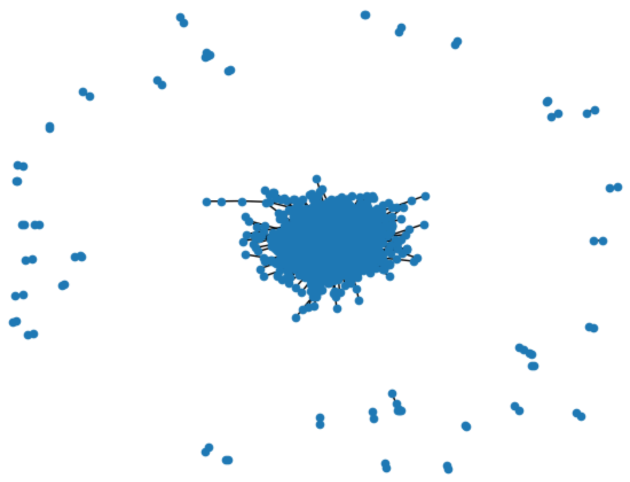
Modularity is a measure of the strength of division of a network into communities or modules, with higher values indicating a better community structure.

We have also noted the time taken by each algorithm for each edge count and plotted it separately as well as together.

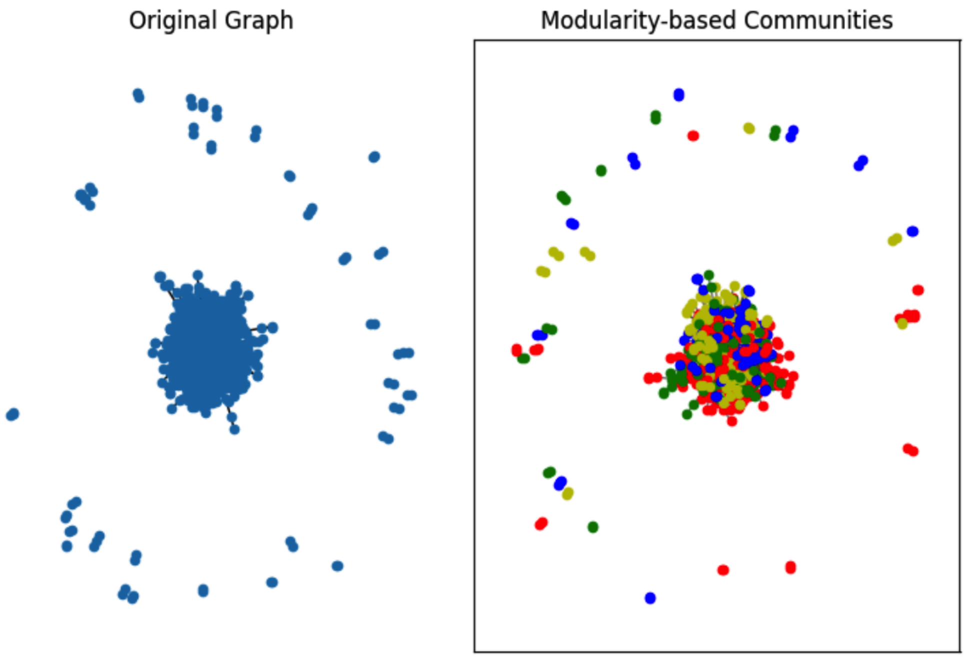
```
t_gn=t.time()
girvan_newman_communities = girvan_newman(G)
print("Time taken by the Girvan-Newman algorithm = ", t.time()-t_gn, " seconds")
t_lp=t.time()
label_propagation_communities = label_propagation(G)
print("Time taken by the Label Propagation algorithm = ", t.time()-t_lp, " seconds")
t_mod=t.time()
modularity_communities = modularity(G)
print("Time taken by the Modularity algorithm = ", t.time()-t_mod, " seconds")
```

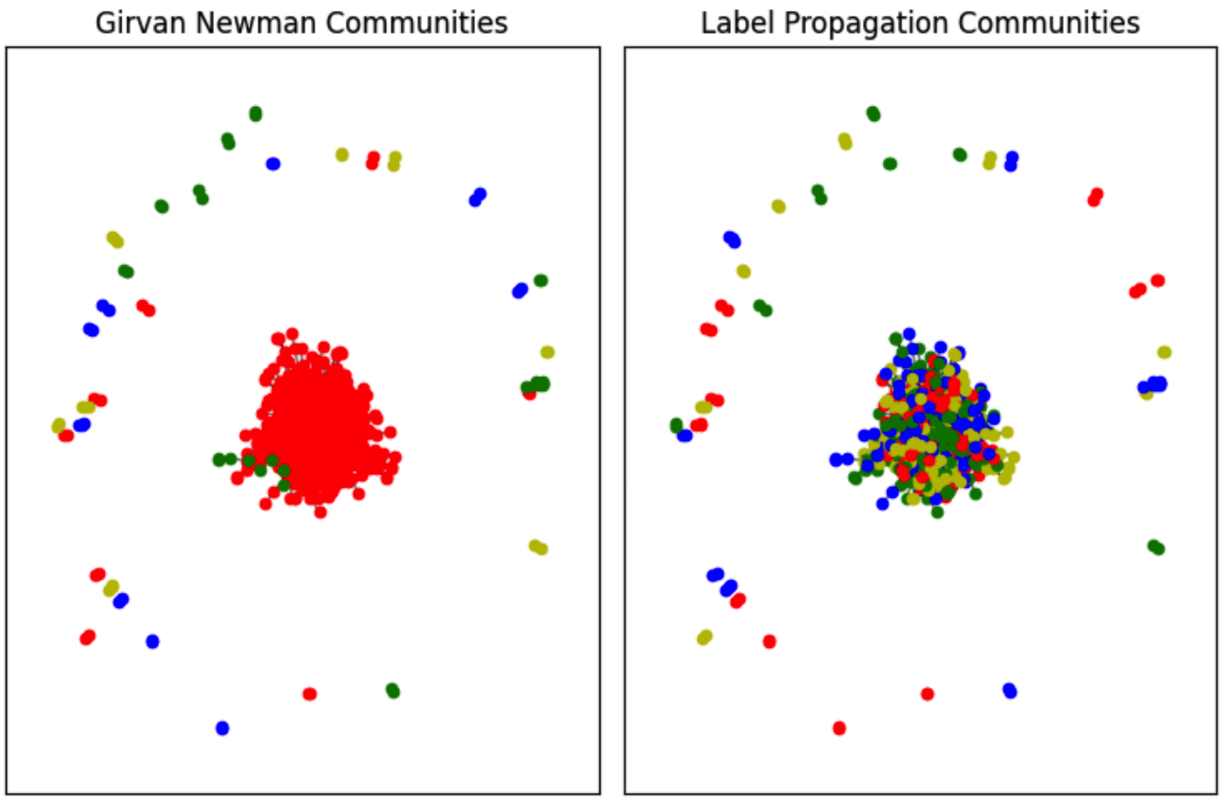
**<u>For 7000 edges:</u>**

Original graph:



Communities detected:



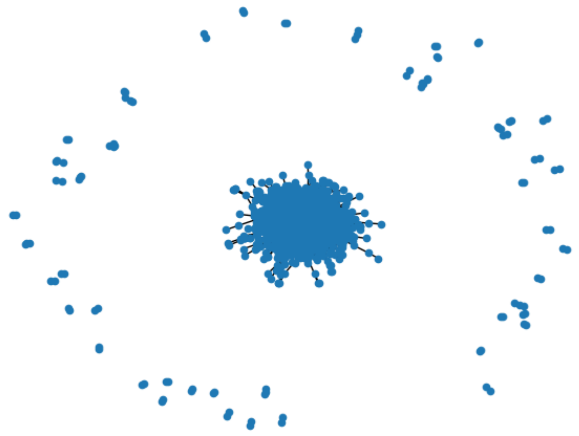


Time taken:

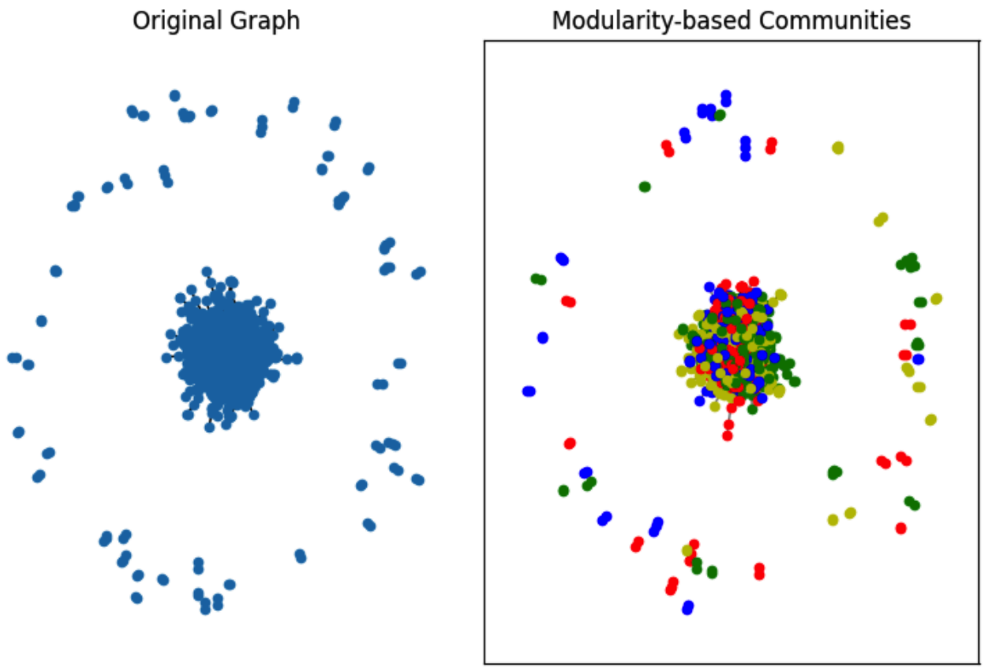
```
Time taken by the Girvan–Newman algorithm =  1274.3662445545197  seconds
Time taken by the Label Propagation algorithm =  0.15946269035339355  seconds
Time taken by the Modularity algorithm =  6.751706600189209  seconds
```

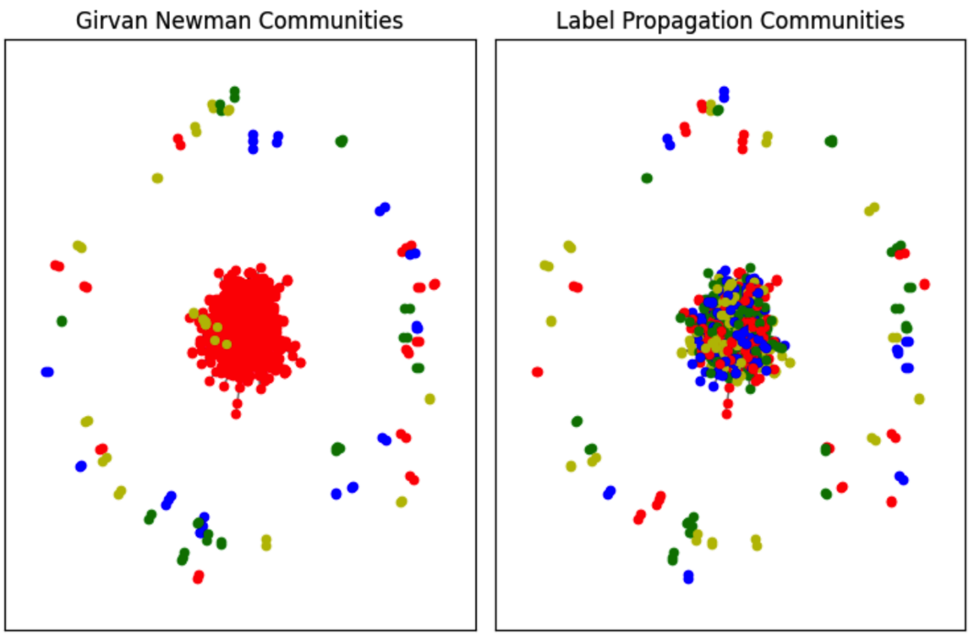
**For 6000 edges:**

Original graph:



Communities detected:

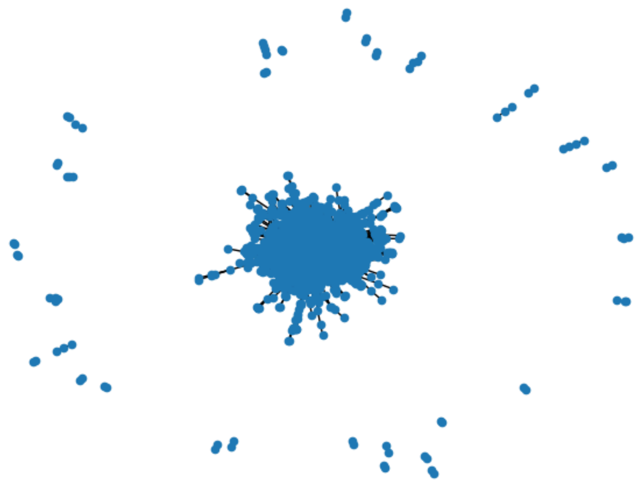




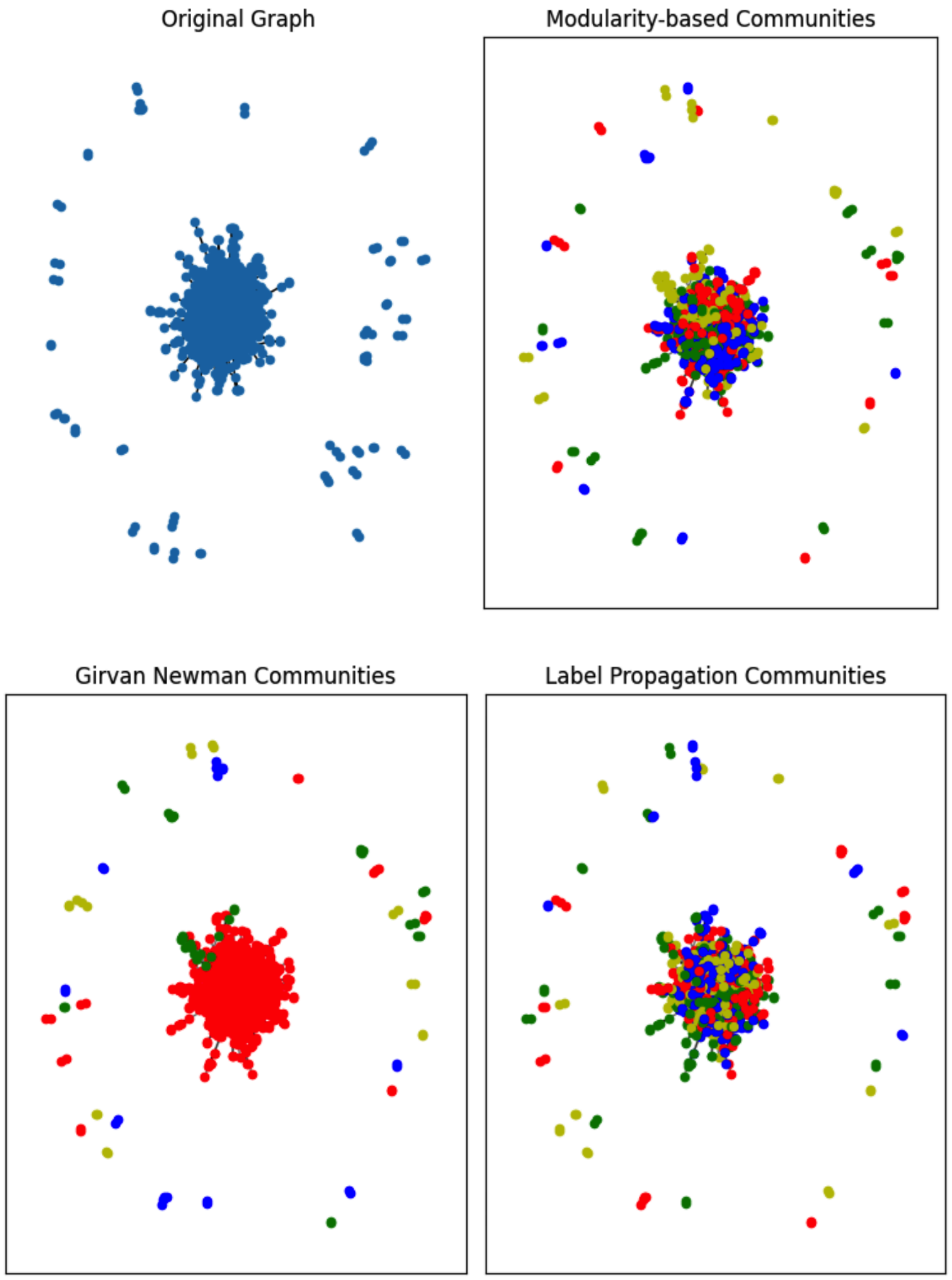
Time taken:

```
Time taken by the Girvan–Newman algorithm =  328.2793753147125  seconds
Time taken by the Label Propagation algorithm =  0.1362013816833496  seconds
Time taken by the Modularity algorithm =  8.40793776512146  seconds
```

**<u>For 5000 edges:</u>**
Original graph:



Communities detected:

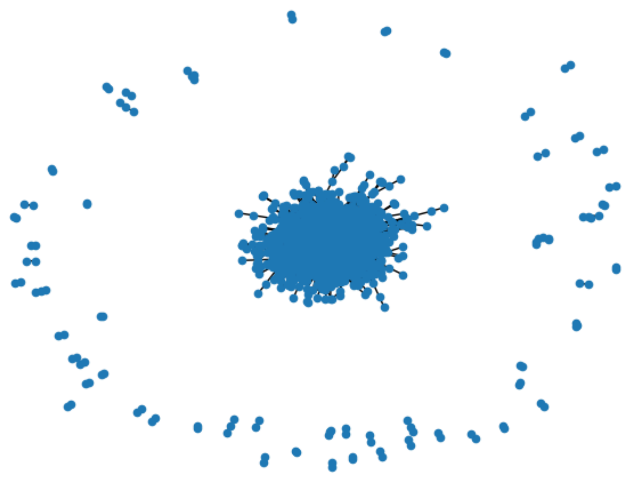

Original Graph
Modularity-based Communities
Girvan Newman Communities
Label Propagation Communities

Time taken:

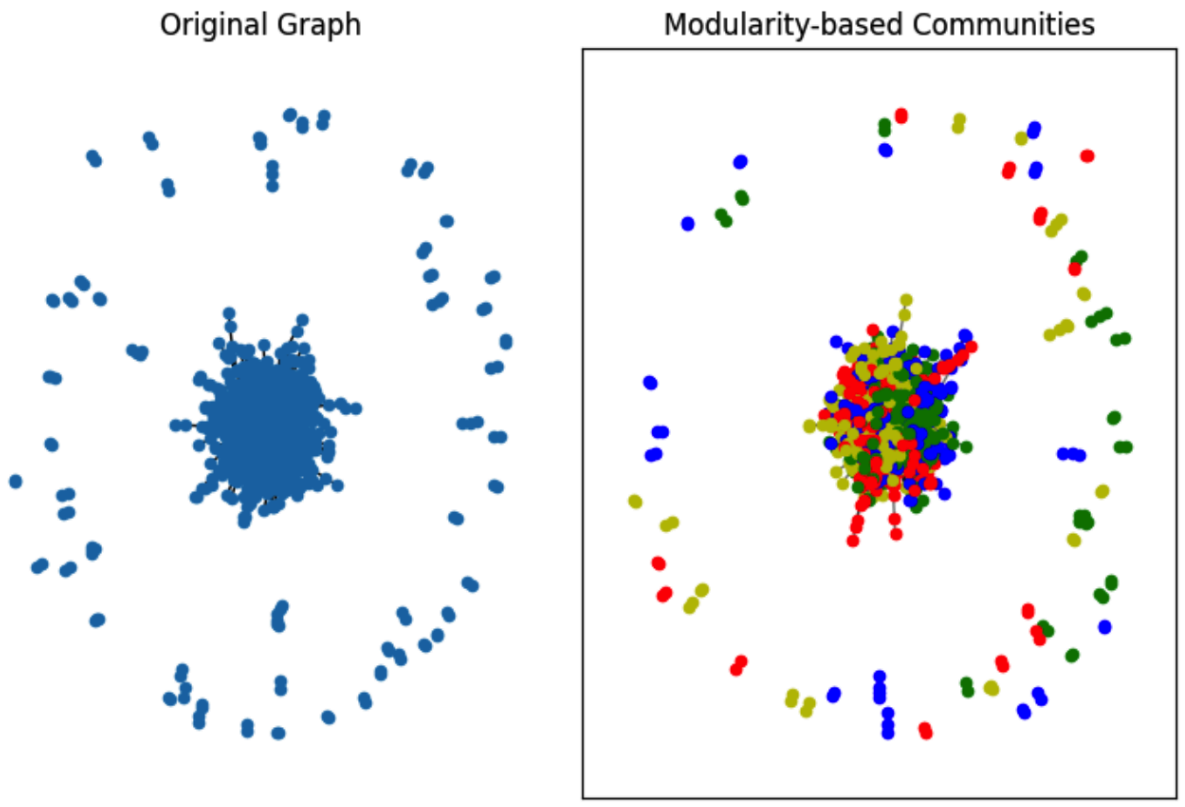
```
Time taken by the Girvan–Newman algorithm =  135.9370768070221  seconds
Time taken by the Label Propagation algorithm =  0.09103083610534668  seconds
Time taken by the Modularity algorithm =  4.495419979095459  seconds
```

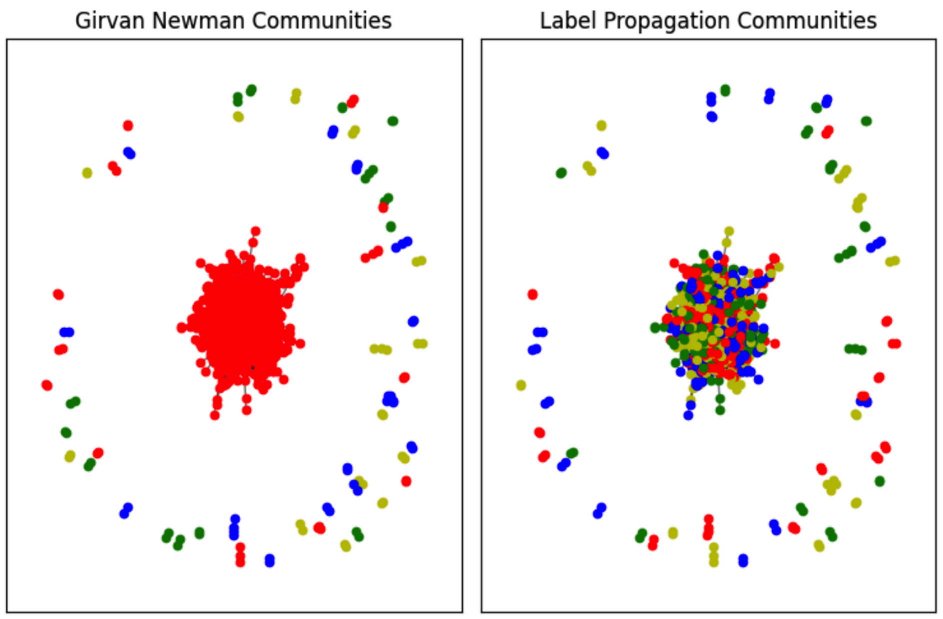
**<u>For 4000 edges:</u>**

Original graph:



Communities detected:



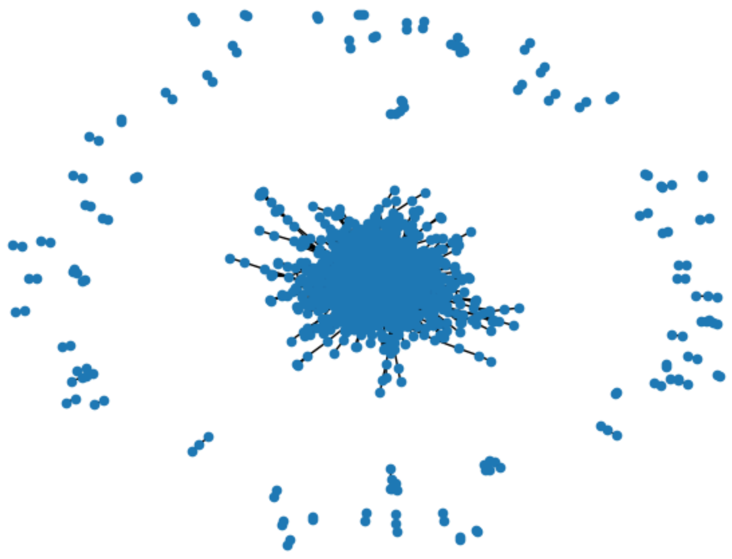


Time taken:

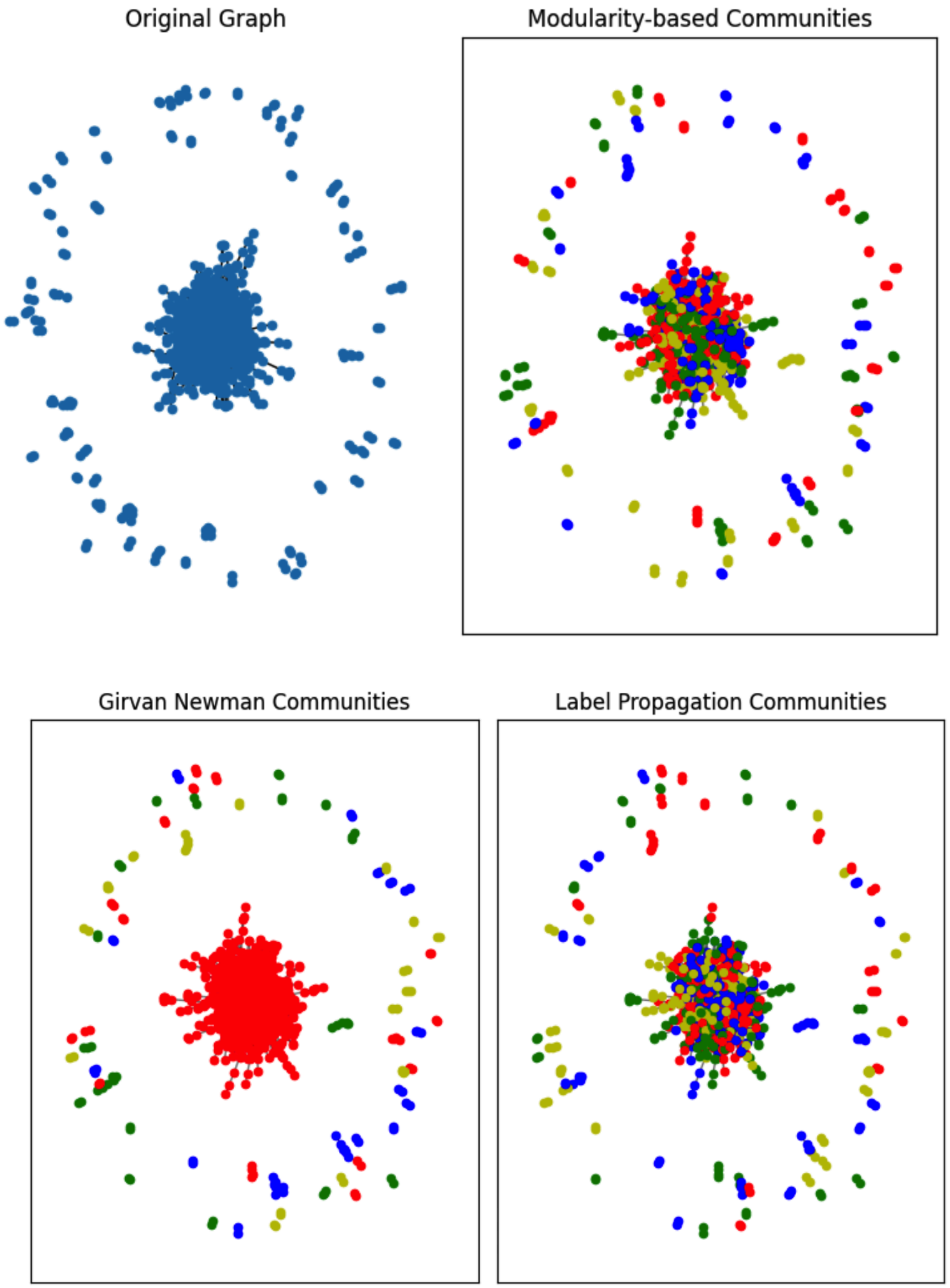
```
 Time taken by the Girvan-Newman algorithm =  422.6694588661194  seconds
 Time taken by the Label Propagation algorithm =  0.057338714599609375  seconds
 Time taken by the Modularity algorithm =  2.84255313873291  seconds
```

**<u>For 3000 edges:</u>**

Original graph:



Communities detected:

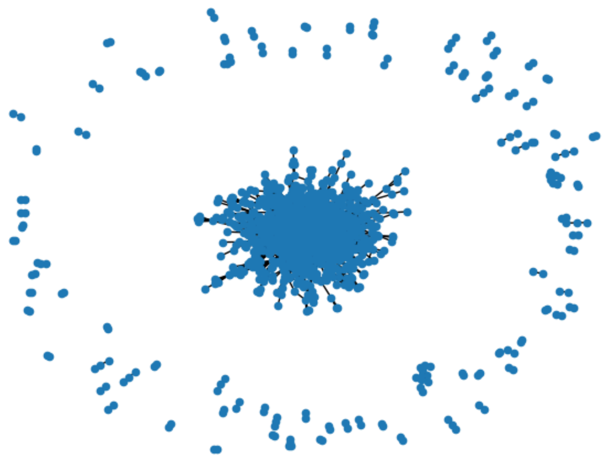


Time taken:

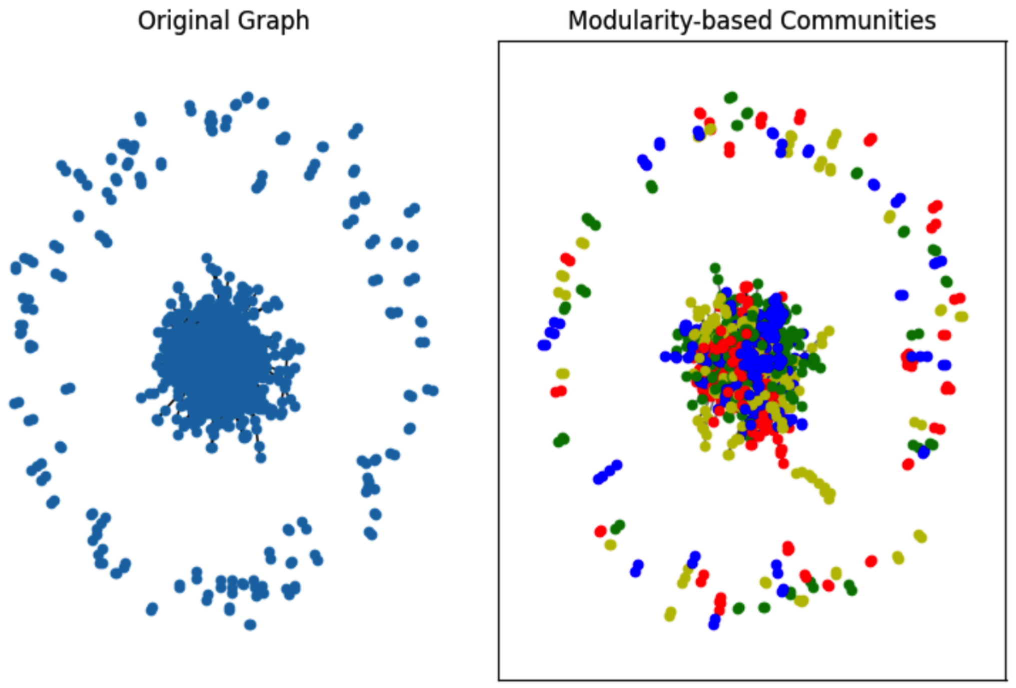
```
Time taken by the Girvan—Newman algorithm =  188.9365108013153  seconds
Time taken by the Label Propagation algorithm =  0.06389093399047852  seconds
Time taken by the Modularity algorithm =  1.7712922096252441  seconds
```

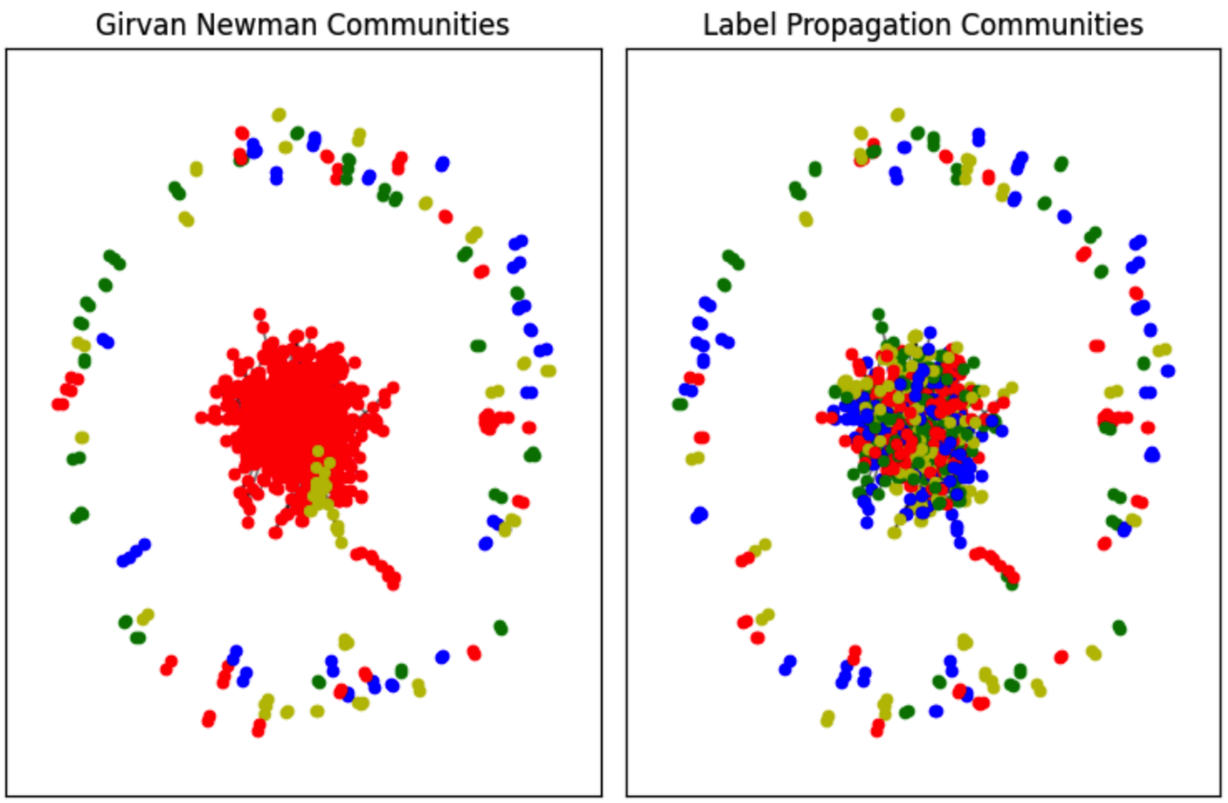
**<u>For 2000 edges:</u>**

Original graph:



Communities detected:



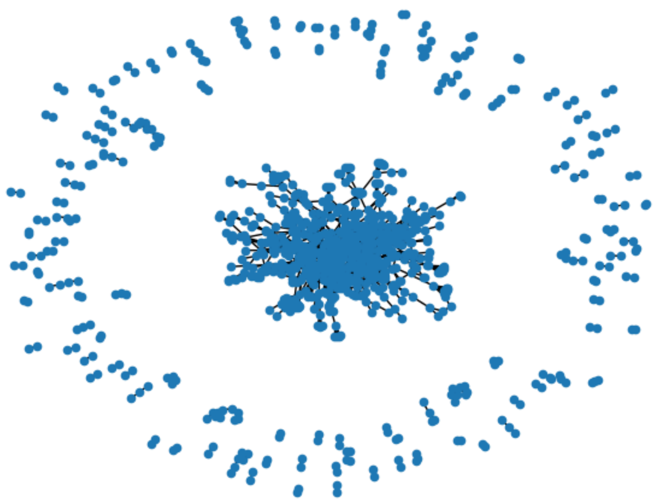


TIme taken:

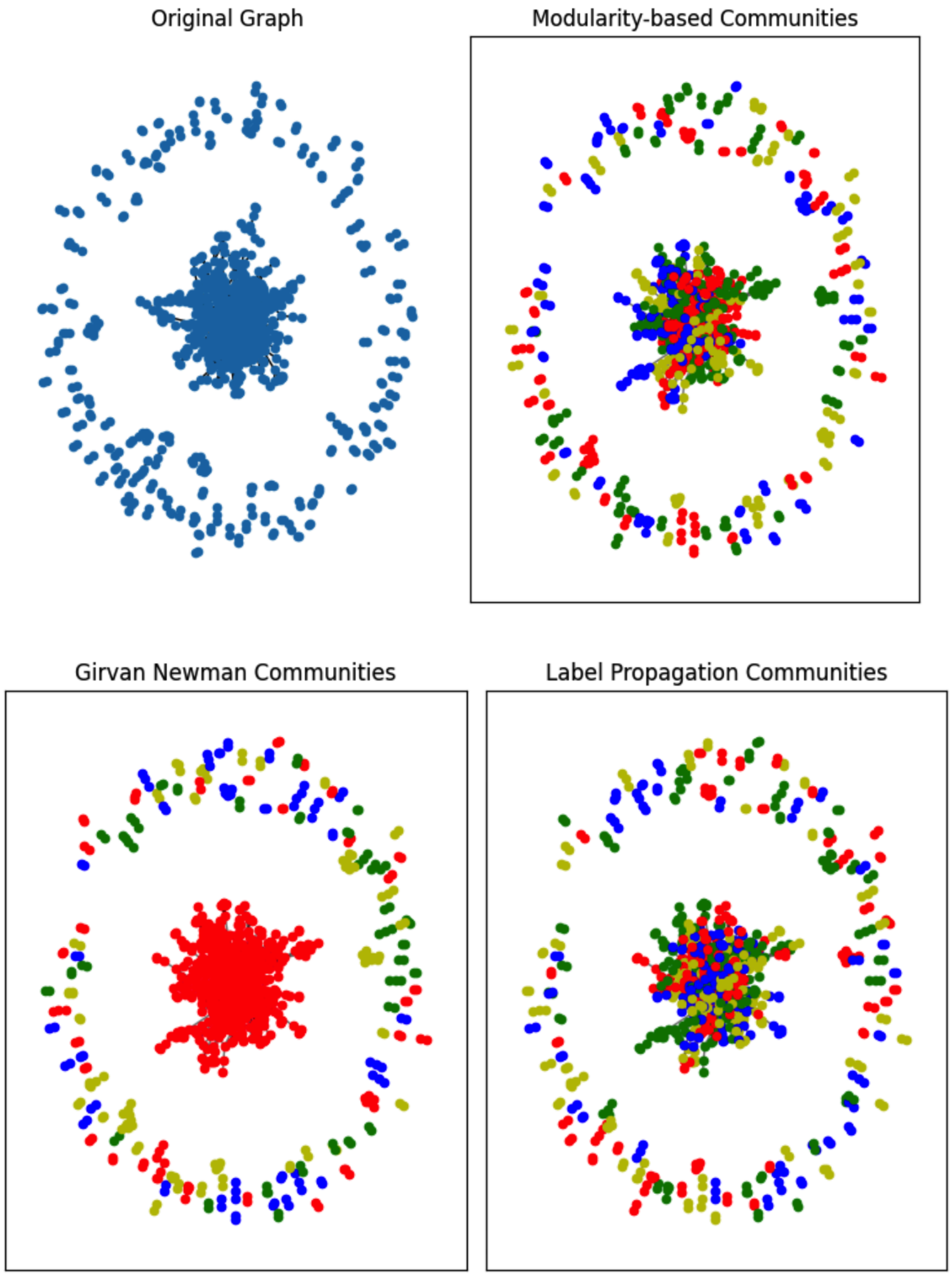
```
Time taken by the Girvan-Newman algorithm =  185.03327775001526  seconds
Time taken by the Label Propagation algorithm =  0.02745962142944336  seconds
Time taken by the Modularity algorithm =  0.5579111576080322  seconds
```

**<u>For 1000 edges:</u>**

Original graph:



Communities detected:

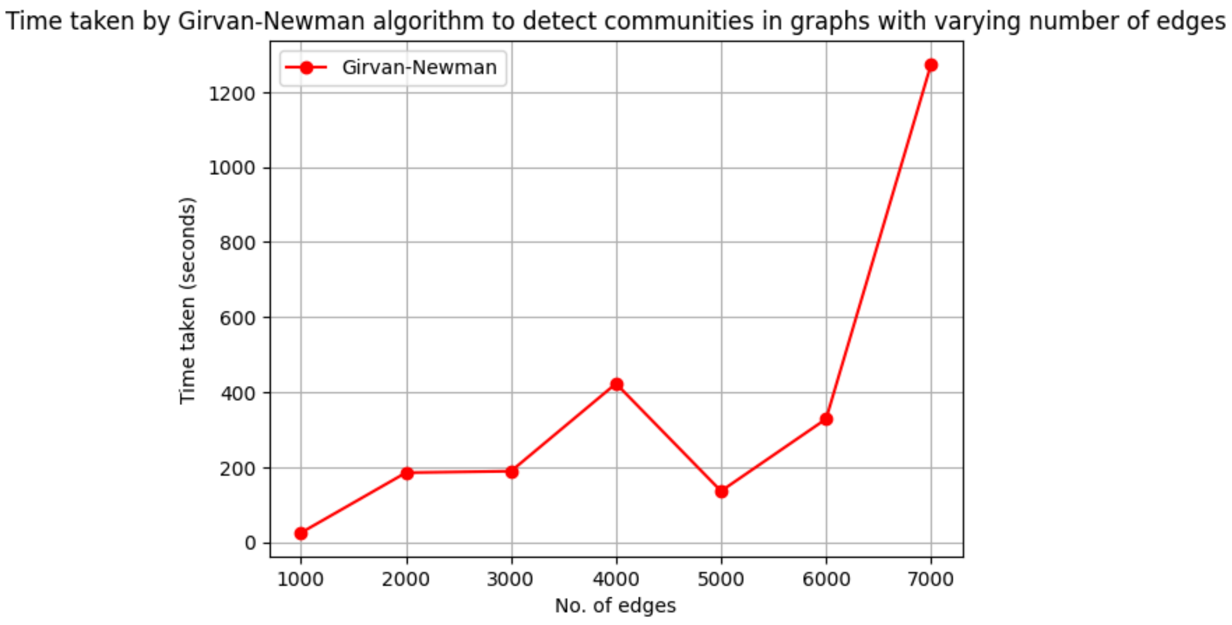


Time taken:

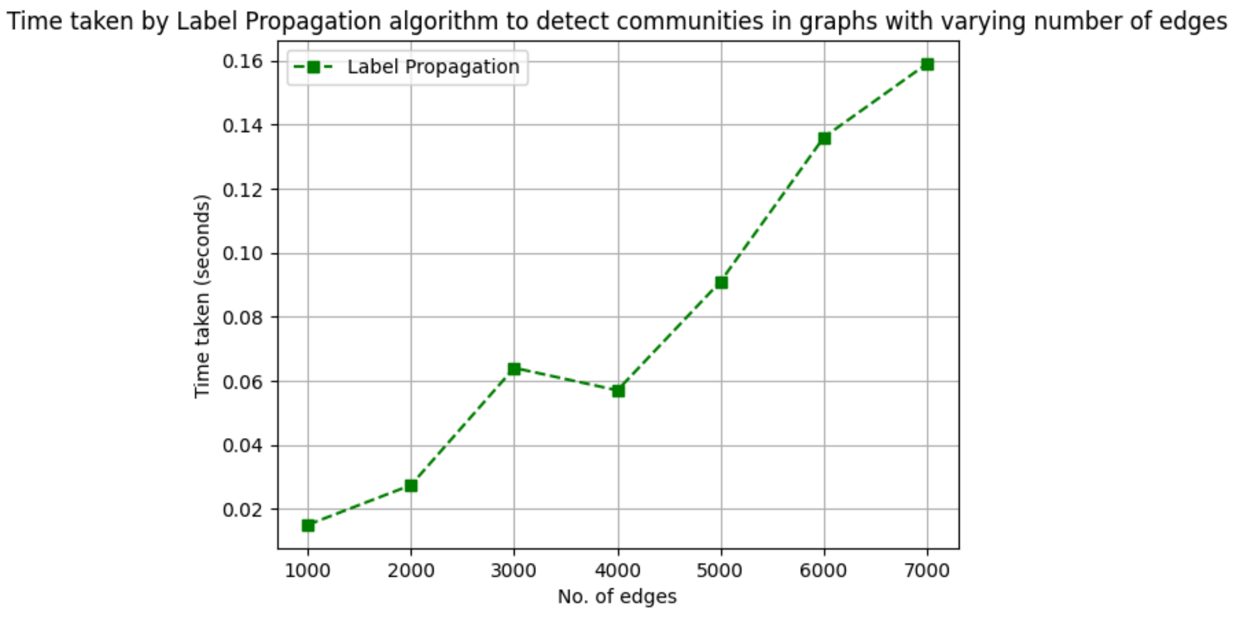
```
Time taken by the Girvan–Newman algorithm =  24.732447385787964  seconds
Time taken by the Label Propagation algorithm =  0.015288114547729492  seconds
Time taken by the Modularity algorithm =  0.1297602653503418  seconds
```

We now plot the graph of time taken by algorithm vs the number of edges.

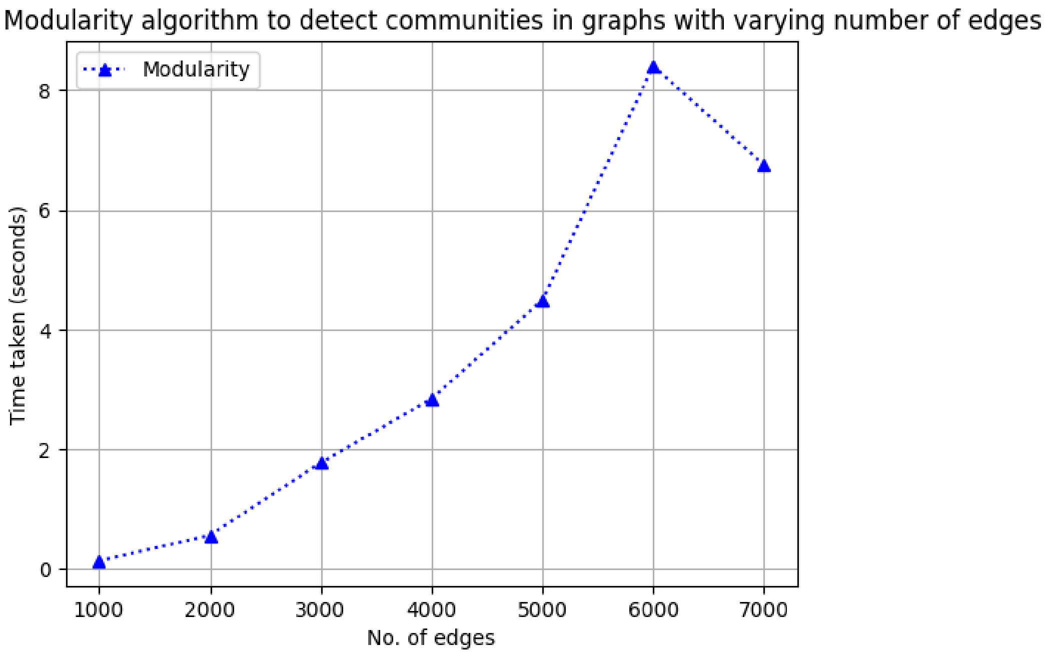
**The Girvan-Newman Algorithm:**



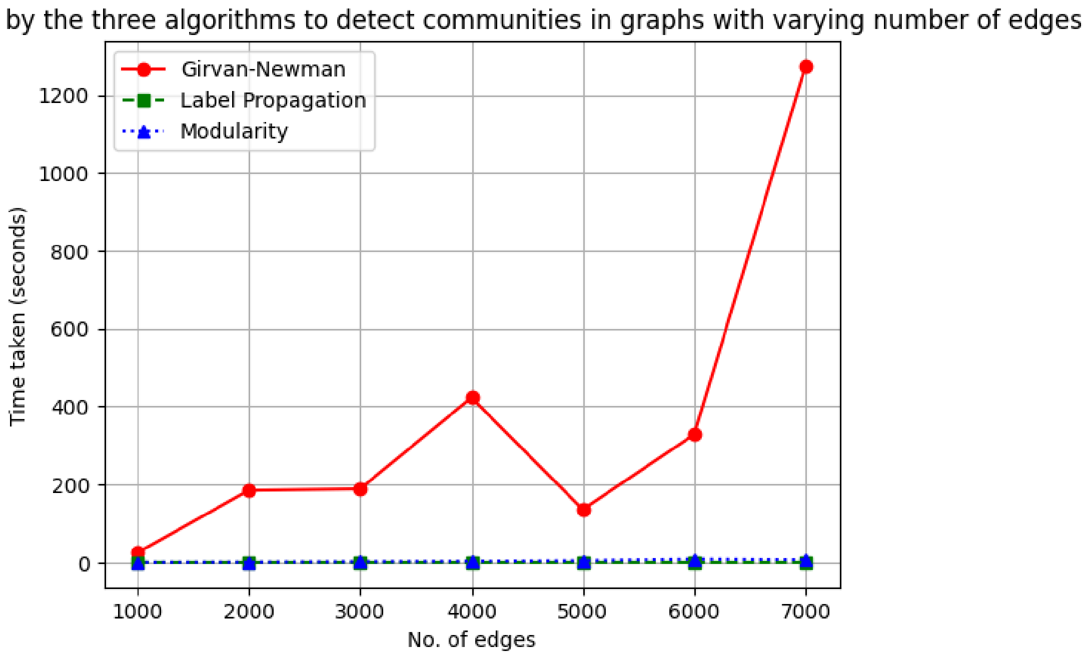
**The Label Propagation Algorithm:**



## The Modularity Community detection algorithm:



Plotting them all together for comparison:



**Conclusion:**

1. For 1000 edges, the three algorithms take very less time. Even then, the Girvan Newman algorithm (~24 secs) takes more time than the other two algorithms (<1 secs).
2. As the number of edges increases from 1000 to 2000, we see a drastic change in the time taken by the Girvan-Newman algorithm, where the other two are much faster and similar to each other.
3. When we compare the Label Propagation algorithm and the Modularity community detection algorithm, we find that the Label Propagation algorithm is much faster than the Modularity community detection algorithm.
4. We can see these values as follows:

   ```
   time_taken_gn=[24.732, 185.033, 188.936, 422.669, 135.937,
   328.279, 1274.367]
   time_taken_lp=[0.015, 0.0274, 0.064, 0.057, 0.091, 0.136,
   0.159]
   time_taken_mod=[0.130, 0.558, 1.771, 2.842, 4.493, 8.408,
   6.752]
   ```

5. As for the quality of the communities detected, the Label Propagation algorithm and the Modularity community detection algorithm give us scattered clusters rather than distinct communities.
6. The Girvan Newman algorithm gives us a more distinct community in the center.
7. We calculated the NMI score of the Label propagation algorithm and the Girvan Newman algorithm taking the communities detected by the Modularity community detection algorithm as the ground truth communities. We did this for the graph with 7000 edges.

   ```
   NMI (Girvan-Newman): 0.17275484996124674
   NMI (Label Propagation): 0.48640638249731466
   ```

   The NMI facilitates comparisons between two clusters or communities, yielding a value that ranges from 0 to 1. A higher value indicates a greater degree of similarity between two partitions or communities.

   This indicated that the Label Propagation communities are much closer to the Modularity community detection communities than the Girvan-Newman communities.
8. Hence, while we find that the communities detected by the Girvan-Newman algorithm seem better, time complexity wise the Label Propagation algorithm proves to be the best.