NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. VIth Sem

Subject Code: CS-321

Subject Name: Social Network Analysis Lab

Submitted By:

Name	: Subhojit Ghimire
Sch. Id.	: 1912160
Branch	: CSE – B

AIM: TO GENERATE THE FOLLOWING MENTIONED RANDOM GRAPHS AND ANALYSE THE PROS AND CONS IN THE CONTEXT OF LARGE-SCALE GRAPHS.

GRAPH SETS: **i. Erdos-Renyi Model (using network library)**

ii. Albert-Barabasi Model (using network library)

iii. LFR Graph (using LFR graph source code)

THEORY:

1. **Erdos-Renyi Model:** This model is used for generating random graphs. This model can be used in the probabilistic method to prove the existence of graphs satisfying various properties. In Erdos-Renyi network, we assign N nodes and then connect each pair with probability p. This means that no one node will have much higher degree than any other.

2. **Albert-Barabsai Model:** This model is used for generating random scale-free network with power-law degree distribution. This model tries to explain the existence of such nodes in real network which are thought to be approximately scale-free but contain few nodes with unusually high degree as compared to the other nodes of the network. In Barabsai-Albert network, we assign N nodes, but to create them, we first start with a small set of connected nodes. Then we add nodes one at a time till we get N nodes.

3. **LFR** (**Lancichinetti–Fortunato–Radicchi benchmark) Graph:** The LFR algorithm is used to generate benchmark networks, i.e., artificial networks that resemble real-world networks. They have a priori known communities and are used to compare different community detection methods. The advantage of Lancichinetti–Fortunato–Radicchi benchmark over other methods is that it accounts for the heterogeneity in the distributions of node degrees and of community sizes.

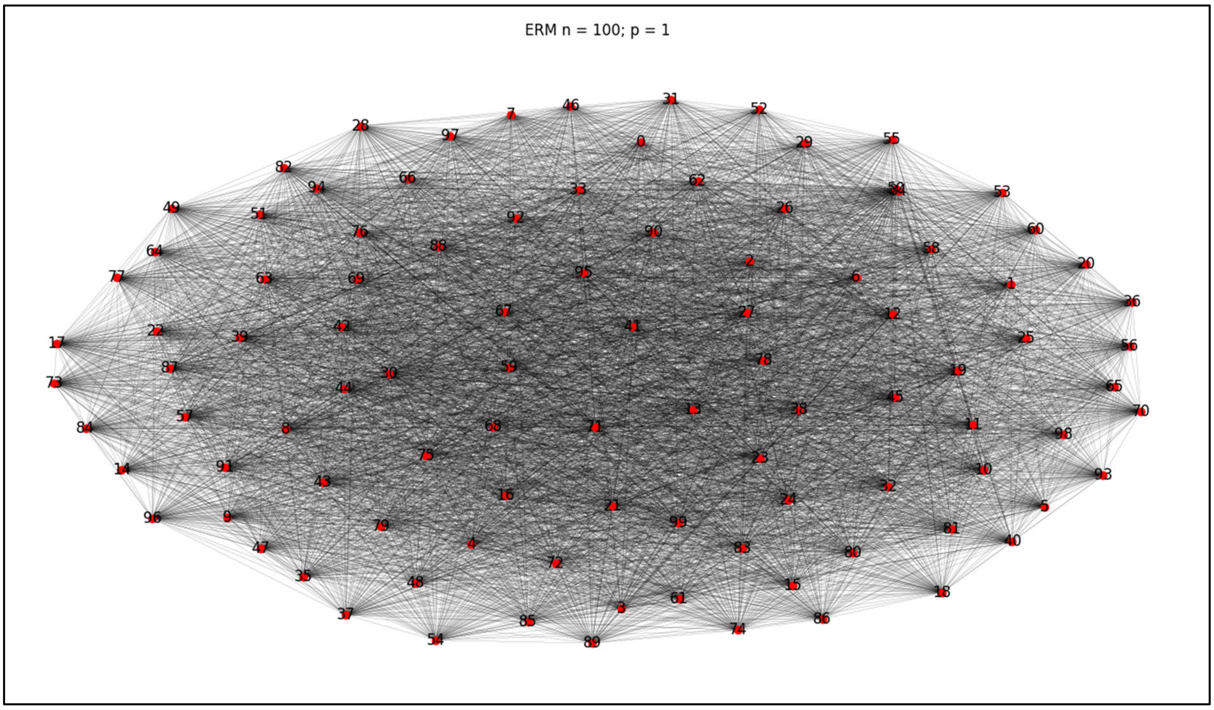
CODE:

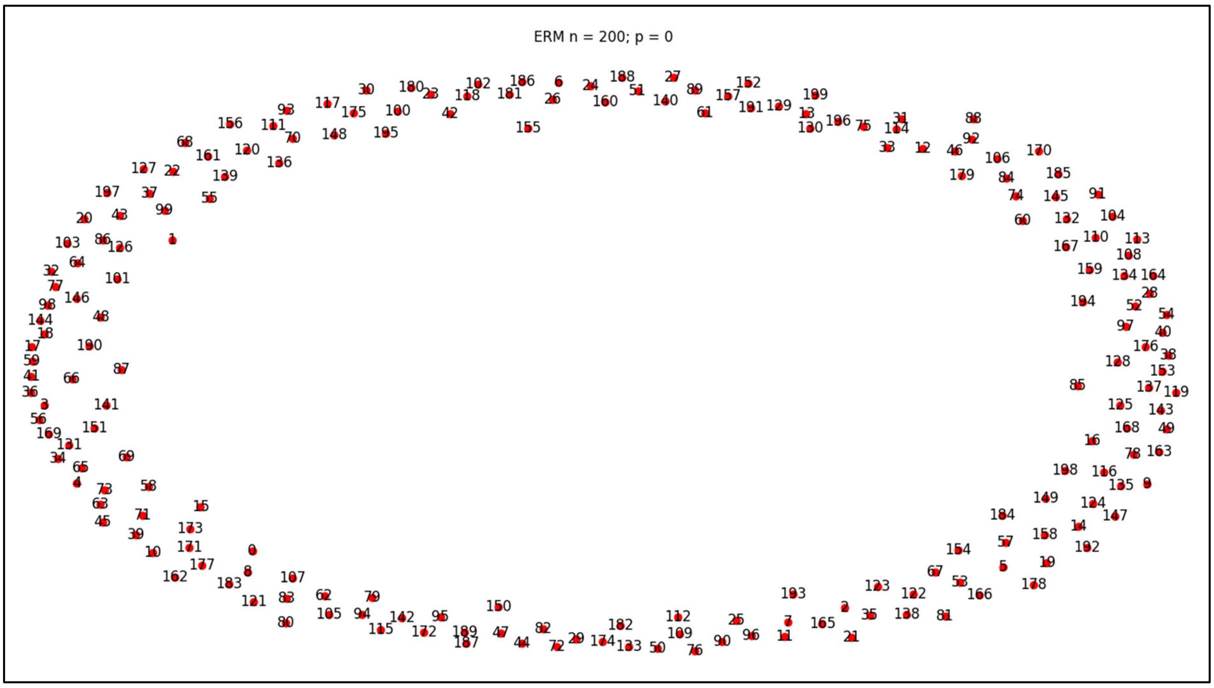
```
import matplotlib.pyplot as plt
import networkx as nx
from networkx.algorithms.community import LFR_benchmark_graph

# Graph Node Setting
options = {
    "node_color": "red",
    "node_size": 50,
    "linewidths": 0,
    "width": 0.1,
}

# ERDOS-RENYI MODEL
ERM_1 = nx.erdos_renyi_graph (100, 0)
ERM_2 = nx.erdos_renyi_graph (200, 0.02)
ERM_3 = nx.erdos_renyi_graph (300, 0.2)
ERM_4 = nx.erdos_renyi_graph (400, 1)

plt.figure ()
plt.title ("ERM n = 100; p = 0")
nx.draw (ERM_1, with_labels = True, **options)

plt.figure ()
plt.title ("ERM n = 200; p = 0.02")
nx.draw (ERM_2, with_labels = True, **options)

plt.figure ()
plt.title ("ERM n = 300; p = 0.2")
nx.draw (ERM_3, with_labels = True, **options)

plt.figure ()
plt.title ("ERM n = 400; p = 1")
nx.draw (ERM_4, with_labels = True, **options)

plt.show ()

# ALBERT-BARABASI MODEL
ABM_1 = nx.barabasi_albert_graph (100, 15)
ABM_2 = nx.barabasi_albert_graph (200, 5)
ABM_3 = nx.barabasi_albert_graph (300, 20)
ABM_4 = nx.barabasi_albert_graph (400, 10)

plt.figure ()
plt.title ("ABM n = 100; m = 20")
nx.draw (ABM_1, with_labels = True, **options)
```

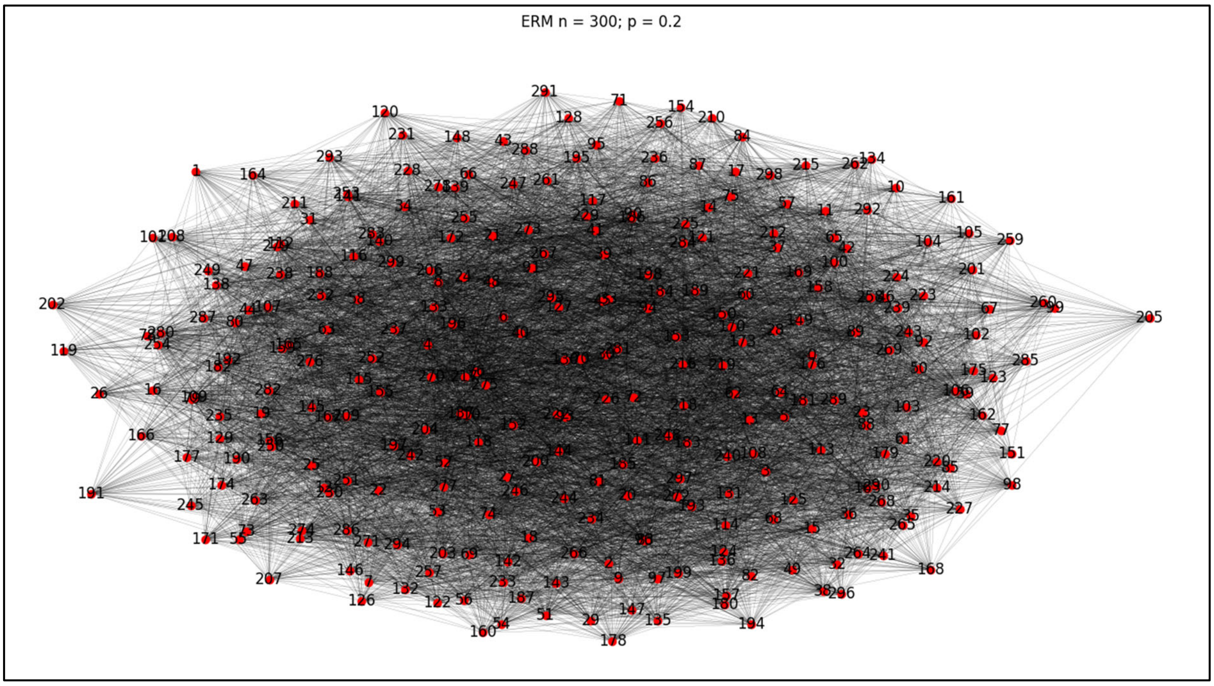
```
plt.figure ()
plt.title ("ABM n = 200; m = 15")
nx.draw (ABM_2, with_labels = True, **options)

plt.figure ()
plt.title ("ABM n = 300; m = 10")
nx.draw (ABM_3, with_labels = True, **options)

plt.figure ()
plt.title ("ABM n = 400; m = 5")
nx.draw (ABM_4, with_labels = True, **options)

plt.show ()

# LANCICHINETTI-FORTUNATO-RADICCHI BENCHMARK
LFR_1 = LFR_benchmark_graph (n = 250, tau1 = 3, tau2 = 1.5, mu = 0.1,
average_degree = 5, min_community = 20, seed = 10)
LFR_2 = LFR_benchmark_graph (n = 250, tau1 = 2, tau2 = 1.1, mu = 0.1,
min_degree = 20, max_degree = 70, max_iters = 1000, seed = 10)
LFR_3 = LFR_benchmark_graph (n = 350, tau1 = 2, tau2 = 1.1, mu = 0.1,
min_degree = 20, max_degree = 50, max_iters = 5000, seed = 20)
LFR_4 = LFR_benchmark_graph (n = 450, tau1 = 2, tau2 = 1.1, mu = 0.1,
min_degree = 20, max_degree = 50, max_iters = 2500, seed = 10)

plt.figure ()
plt.title ("LFR n = 250; tau1 = 3; tau2 = 1.5; mu = 0.1; average_degree = 5;
min_community = 20; seed = 10")
nx.draw (LFR_1, with_labels = True, **options)

plt.figure ()
plt.title ("LFR n = 250; tau1 = 2; tau2 = 1.1; mu = 0.1; min_degree = 20;
max_degree = 50; max_iters = 1000; seed = 10")
nx.draw (LFR_2, with_labels = True, **options)

plt.figure ()
plt.title ("LFR n = 350; tau1 = 2; tau2 = 1.1; mu = 0.1; min_degree = 20;
max_degree = 50; max_iters = 5000; seed = 20")
nx.draw (LFR_3, with_labels = True, **options)

plt.figure ()
plt.title ("LFR n = 450; tau1 = 2; tau2 = 1.1; mu = 0.1; min_degree = 20;
max_degree = 50; max_iters = 2500; seed = 10")
nx.draw (LFR_4, with_labels = True, **options)

plt.show ()
```

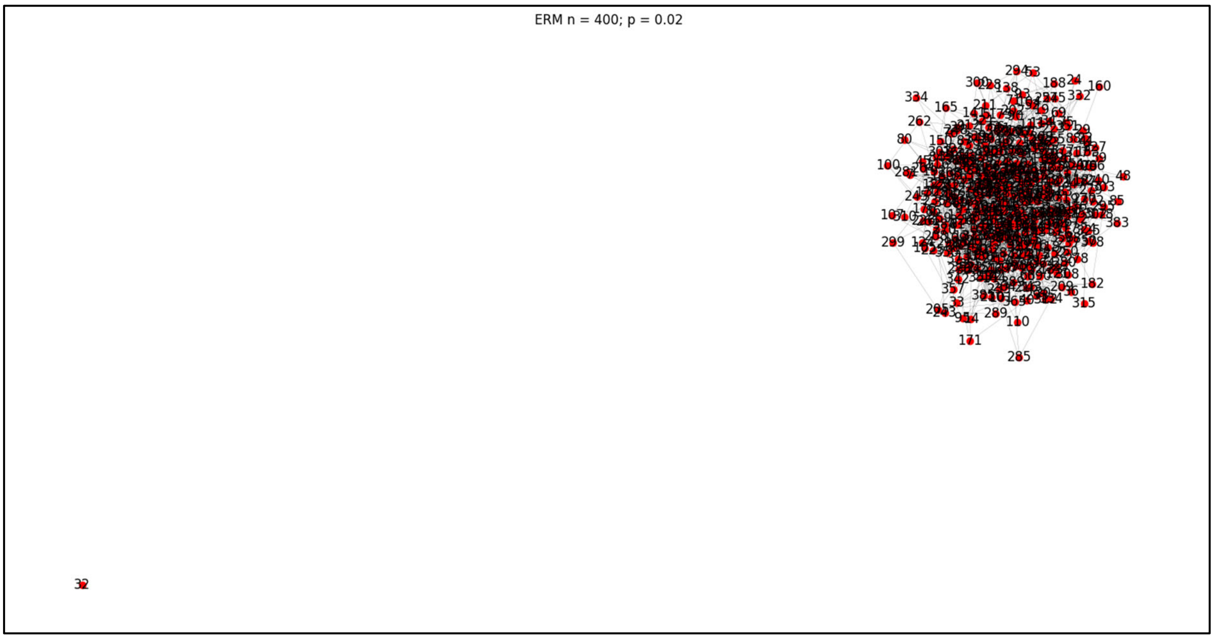
OUTPUT:

```
ERDOS-RENYI MODEL
```

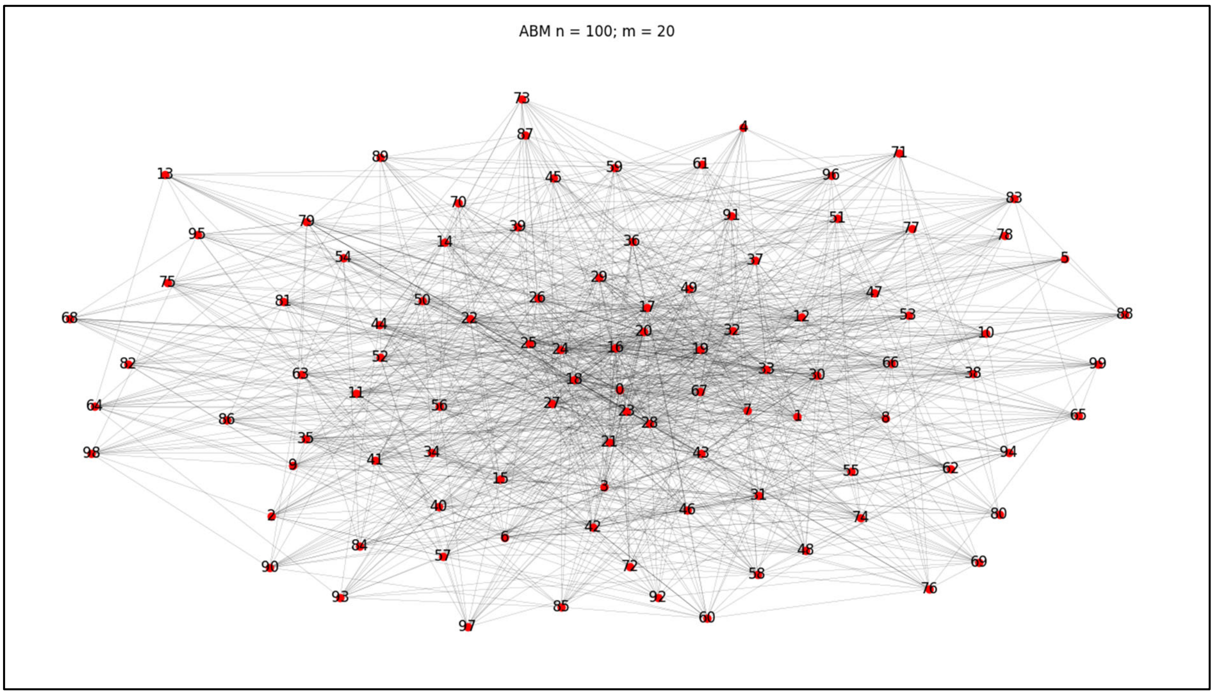




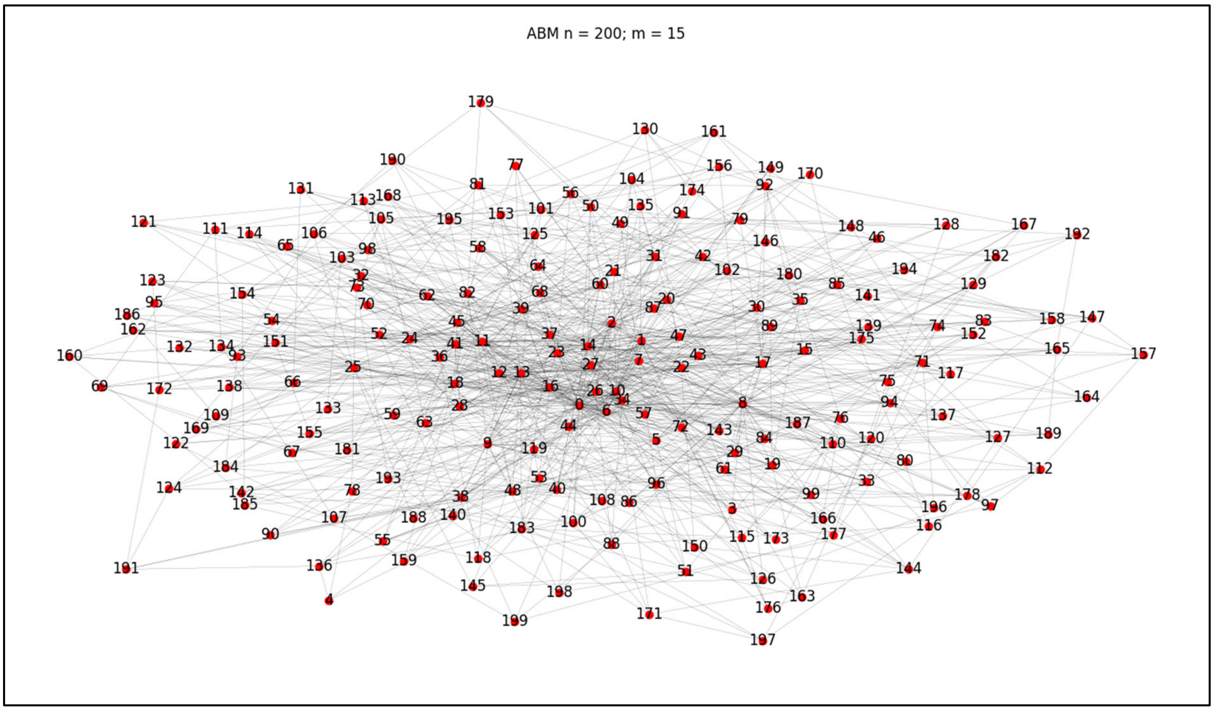

ERM n = 300; p = 0.2


ERM n = 400; p = 0.02

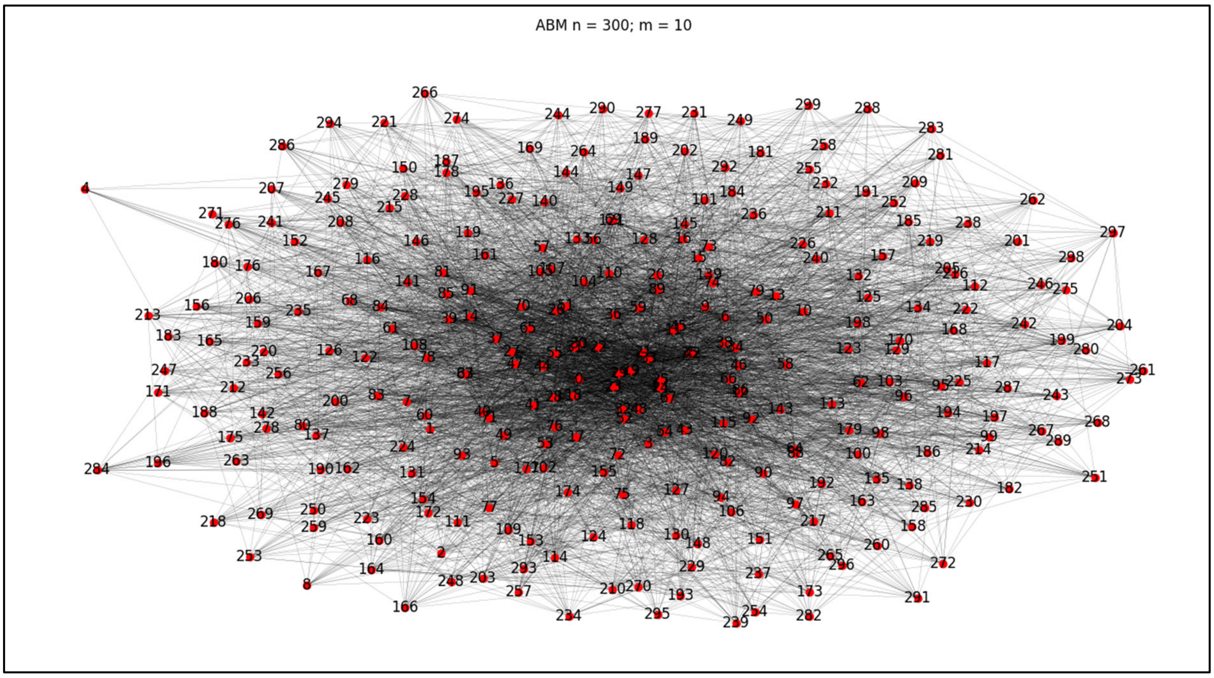
ALBERT-BARABASI MODEL

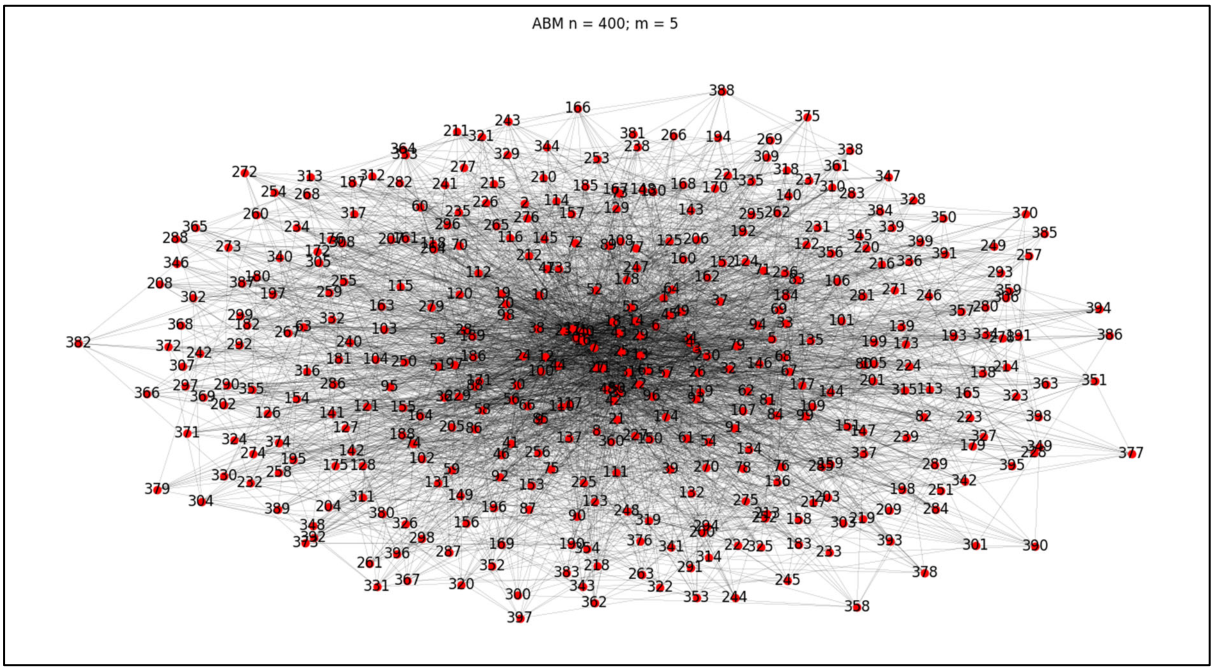


ABM n = 100; m = 20

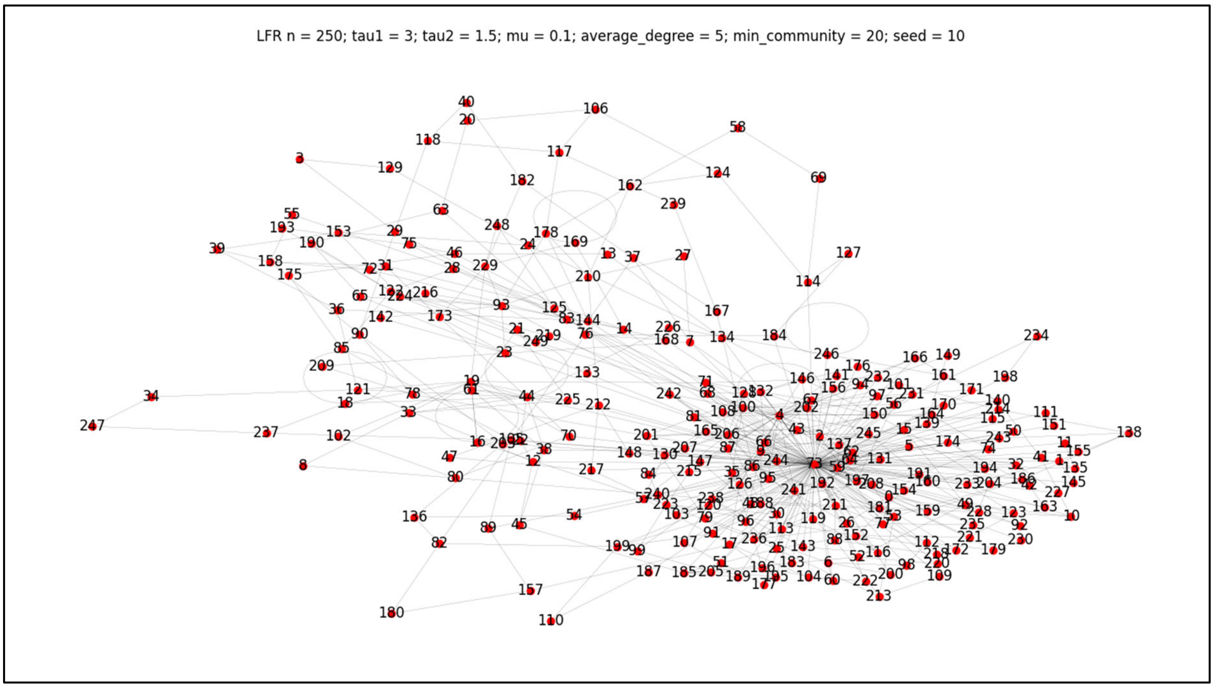


ABM n = 200; m = 15

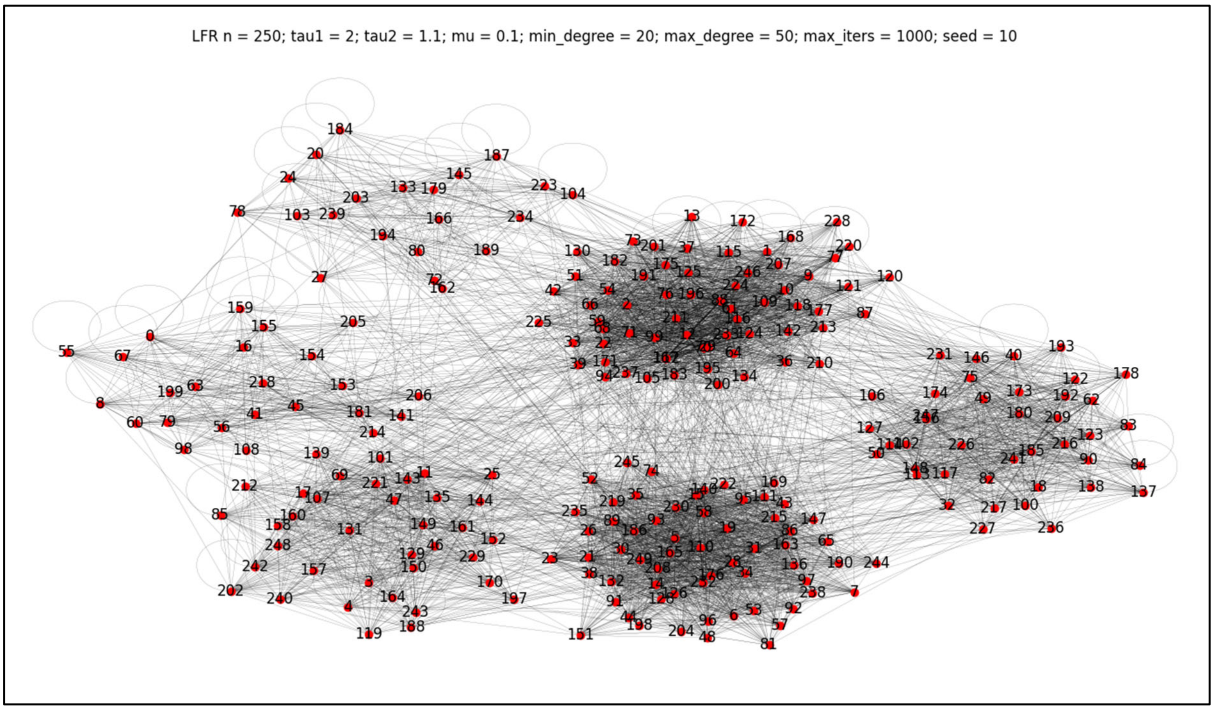

ABM n = 300; m = 10


ABM n = 400; m = 5

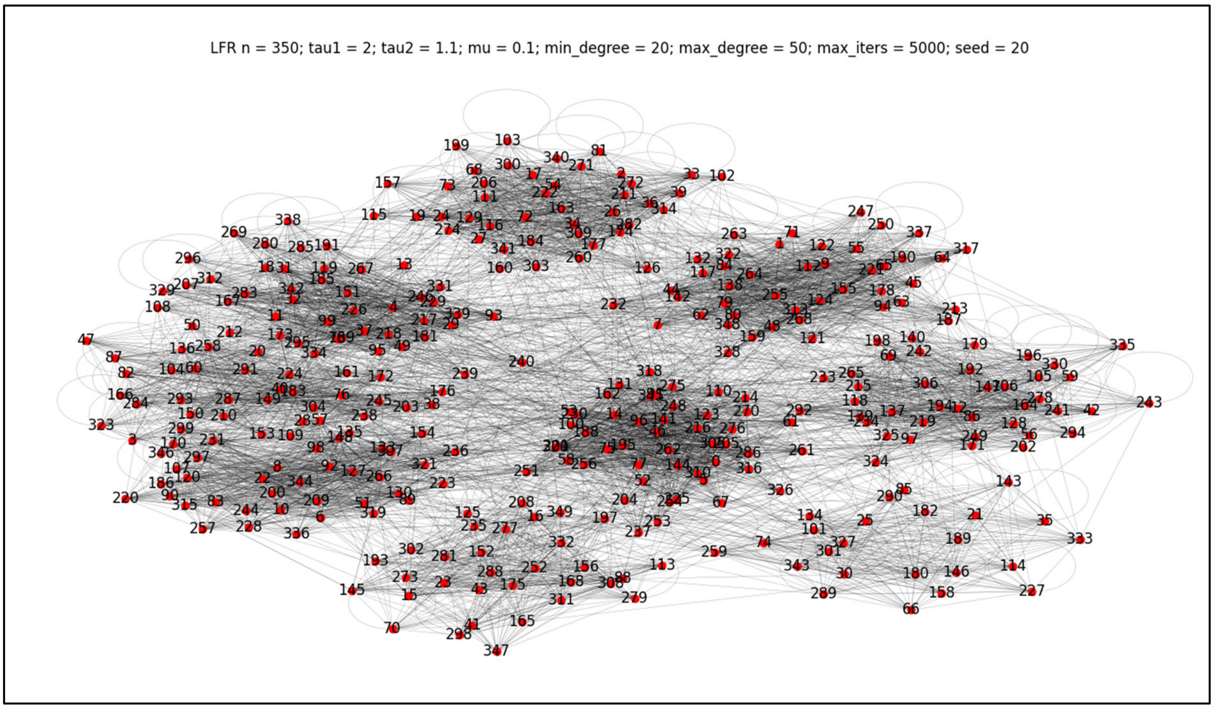
LANCICHINETTI-FORTUNATO-RADICCHI BENCHMARK

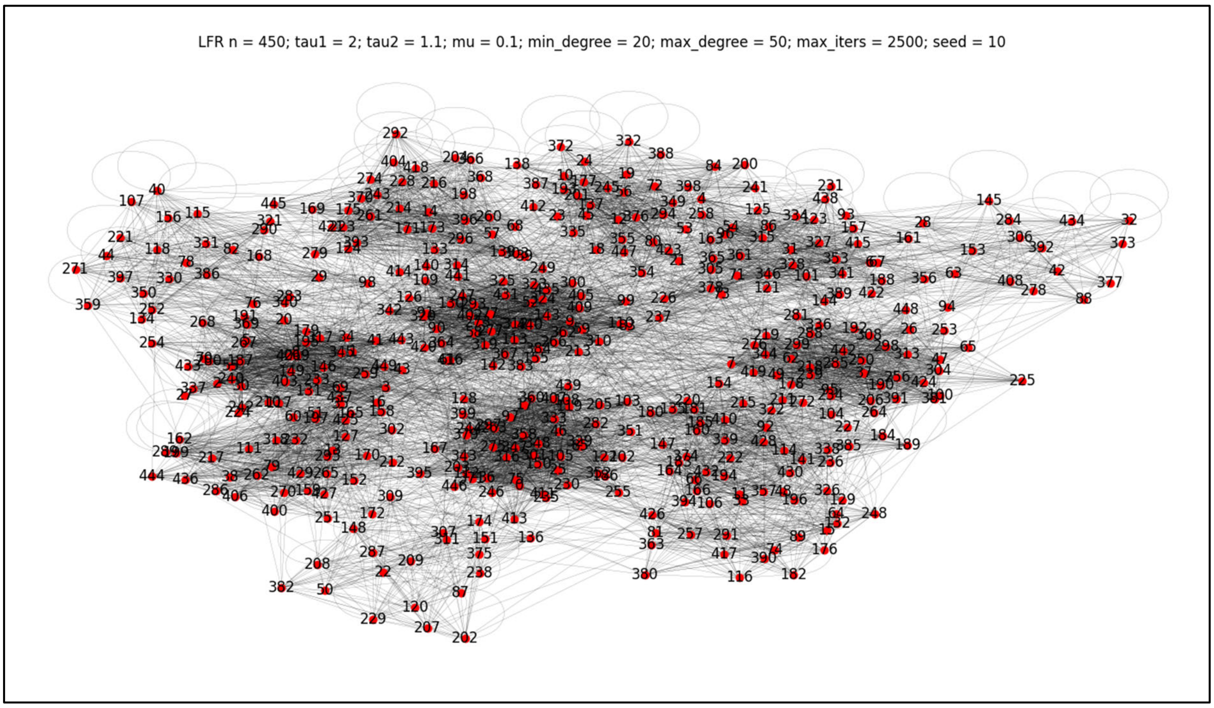


LFR n = 250; tau1 = 3; tau2 = 1.5; mu = 0.1; average_degree = 5; min_community = 20; seed = 10



LFR n = 250; tau1 = 2; tau2 = 1.1; mu = 0.1; min_degree = 20; max_degree = 50; max_iters = 1000; seed = 10


LFR n = 350; tau1 = 2; tau2 = 1.1; mu = 0.1; min_degree = 20; max_degree = 50; max_iters = 5000; seed = 20


LFR n = 450; tau1 = 2; tau2 = 1.1; mu = 0.1; min_degree = 20; max_degree = 50; max_iters = 2500; seed = 10