A black and white image of a building

Description automatically generated with medium confidence

Budapest University of Technology and Economics

Faculty of Electrical Engineering and Informatics

**Modelling Seminar for Engineers**

**Topic:** Network Science - Barabási-Albert

**Written by:** Yagublu Nurlan

**Neptun:** G72RIJ

**Date:** 09.01.2024

Short summary about the paper:

The properties of huge networks, like genetic networks or the World Wide Web, are covered in the paper "Emergence of Scaling in Random Networks," written by Albert-László Barabási and Réka Albert. It introduces the concept of scale-free networks and proposes a model to explain their emergence. This model is based on two key mechanisms: growth and preferential attachment. According to the paper, networks expand continuously by the addition of new vertices, and new vertices attach preferentially to sites that are already well connected.

To implement the model described in the paper and generate a network, we can follow these steps:

1. **Initialization**: Starting with the small number (m0) of vertices.
2. **Growth**: At every time step, add a new vertex
3. **Preferential Attachment**: Each new vertex is connected to m existing vertices with a probability that is proportional to the number of links that the existing vertices already have.
4. **Repeat:** Repeat step 2 and 3 for a large number of steps to grow the network.

In Python, we can implement this model using libraries like NetworkX for network manipulation and Matplotlib for plotting. The code will create a network and calculate its properties such as diameter, clustering coefficient, and degree distribution. Then, we can compare these properties with those of the Internet.

Here is the main.py file for demonstrating plots:

import networkx as nx

import matplotlib.pyplot as plt

import numpy as np

def barabasi\_albert\_graph(n, m0, m):

    """Generates a network using the Barabási-Albert (BA) model.

    Parameters:

    n (int): Total number of nodes in the graph.

    m0 (int): Initial number of nodes.

    m (int): Number of edges to attach from a new node to existing nodes.

    Returns:

    G (networkx.Graph): The generated graph.

    """

    # Initialize a graph with m0 nodes

    G = nx.complete\_graph(m0)

    for node in range(m0, n):

        # For each new node, connect it to m existing nodes

        # The probability of connecting to a node is proportional to its degree

        targets = \_select\_targets(G, m)

        G.add\_node(node)

        for target in targets:

            G.add\_edge(node, target)

    return G

def \_select\_targets(G, m):

    """Selects target nodes based on preferential attachment.

    Parameters:

    G (networkx.Graph): The current graph.

    m (int): Number of targets to select.

    Returns:

    targets (list): List of selected target nodes.

    """

    # Compute the sum of degrees

    degree\_sum = sum(dict(G.degree()).values())

    # Probabilities for each node

    probabilities = [G.degree(node) / degree\_sum for node in G.nodes()]

    # Choose nodes based on their probability

    targets = np.random.choice(G.nodes(), size=m, replace=False, p=probabilities)

    return targets

# Parameters

n = 200   # Total number of nodes

m0 = 5   # Initial number of nodes

m = 3    # Number of edges to attach from a new node to existing nodes

# Generate the graph

ba\_graph = barabasi\_albert\_graph(n, m0, m)

# Plot the graph

plt.figure(figsize=(12, 8))

nx.draw(ba\_graph, node\_size=50, node\_color='blue', with\_labels=False)

plt.title("Barabási-Albert Model Graph")

plt.show()

The generated graph with 200 nodes is visualized below:

A network of lines and dots

Description automatically generated

Next, I will calculate and analyze the properties of this network, specifically its diameter, clustering coefficient, and degree distribution. After that, I'll compare these properties with those known for the Internet.

Comparison.py file:

import networkx as nx

ba\_graph = nx.barabasi\_albert\_graph(200, 5, 3)  # Example parameters (n=200 nodes, m0=5 Initial number of nodes, m = 3  # Number of edges to attach from a new node to existing nodes)

def calculate\_network\_properties(G):

    """Calculate various properties of the network.

    Parameters:

    G (networkx.Graph): The graph for which to calculate properties.

    Returns:

    properties (dict): Dictionary containing network properties.

    """

    properties = {}

    # Diameter: Longest shortest path in the network

    if nx.is\_connected(G):

        properties['diameter'] = nx.diameter(G)

    else:

        properties['diameter'] = "undefined (disconnected graph)"

    # Clustering Coefficient: Measure of degree to which nodes tend to cluster together

    properties['average\_clustering\_coefficient'] = nx.average\_clustering(G)

    # Degree Distribution: Probability distribution of the degrees over the network

    degrees = [G.degree(n) for n in G.nodes()]

    properties['degree\_distribution'] = degrees

    return properties

# Calculate network properties for the BA model graph

ba\_properties = calculate\_network\_properties(ba\_graph)

# Output the calculated properties

print(ba\_properties)

Here are the resultA screenshot of a computer

Description automatically generated

The properties of the generated Barabási-Albert network with 200 nodes are as follows:

1. **Diameter**: The diameter of the network, which is the longest shortest path between any two nodes, is 4. This suggests a small-world property, where most nodes can be reached from any other in a small number of steps.

The Internet also exhibits small-world properties, with a relatively small diameter. However, the exact diameter of the Internet can vary depending on the scale and the type of network being analyzed (e.g., the router level, the autonomous system level). Generally, the Internet's diameter is considered to be quite small, which aligns with the BA model's properties.

1. **Avarage Clustering Coefficient**: This value is approximately 0.153748. This indicates a low tendency for nodes to form tightly knit groups.

The clustering coefficient of the Internet is typically higher than that of the BA model. In real-world networks, especially social networks, there's a stronger tendency for clustering.

1. **Degree Distribution**: The degree distribution in the BA model follows a power law, meaning a few nodes have a very high degree (many connections), while most nodes have a low degree. This is a characteristic of scale-free networks.

The Internet is known to have a scale-free degree distribution. Like the BA model, it has nodes (such as major servers, routers) with significantly higher connectivity compared to most other nodes. This similarity to the BA model reflects the scale-free nature of the Internet.

In conclusion, the scale-free nature and small-world characteristics of the Internet's topology are well captured by the Barabási-Albert model. It might not, however, be able to completely replicate the greater clustering coefficients that are frequently seen in actual networks, such as the Internet.