

# Analiza sieci dróg na terenie uczelni

## Contents

1. Plan eksperymentów analizy wybranych algorytmów generacji grafów .....	1
2. Budowa środowiska .....	3
Tworzenie grafów .....	3
Narzędzie do analizy grafu .....	4
Wywoływanie analizy dla utworzonych grafów .....	5
3. Przeprowadzenie eksperymentów .....	6
Wywołanie skryptu .....	6
Wizualizacja grafów .....	8
4. Opracowanie wyników i wyciąganie wniosków .....	8
Zestawienie wyników do tabeli .....	8
Analiza wartości .....	9
5. Propozycja kierunków dalszych badań .....	9

## 1. Plan eksperymentów analizy wybranych algorytmów generacji grafów

Przedmiotem eksperymentów będzie utworzenie przykładowych grafów na podstawie wybranych algorytmów generowania grafów. W ramach analizy wykorzystane będą narzędzia pozwalające na wyznaczenie podstawowych właściwości grafu, które pozwolą na ocenę jakościową utworzonych grafów, a następnie na porównanie wyników między tymi grafami.

Wykorzystane zostaną następujące algorytmy:

- 1) Erdos-Renyi Graph – Losowy graf, w którym każda para węzłów jest łączona z podanym prawdopodobieństwem.

- 2) Barabasi-Albert Graph – Graf scale-free gdzie nowe węzły łączą się z już dobrze skomunikowanymi węzłami.
- 3) Watts-Strogatz Graph – Graf small-world, który łączy lokalną strukturę z losowym połączeniem do struktur oddalonych.
- 4) Stochastic Block Model (SBM) Graph – Graf dzielony na bloki, gdzie węzły łączą się częściej wewnątrz bloków niż pomiędzy nimi.
- 5) Planted Partition Graph – Graf z predefiniowanym podziałem na społeczności.
- 6) Karate Club Graph – Graf przedstawiający relacje w klubie karate, w którym rysuje się podział na dwie grupy.
- 7) Random Geometric Graph – Graf, w którym węzły są rozmieszczone losowo w przestrzeni, a połączenia tworzone są pomiędzy węzłami znajdującymi się blisko siebie według określonego promienia.
- 8) Powerlaw Cluster Graph – Graf charakteryzujący się rozkładem potęgowym stopni węzłów oraz silnym klastrowaniem.

Do analizy grafów zostaną wyznaczone następujące właściwości:

- 1) Nodes – Liczba wierzchołków
- 2) Edges – Liczba krawędzi
- 3) Average Degree – Średni stopień wierzchołków
- 4) Average Shortest Path – Średnia długość najkrótszych ścieżek między wierzchołkami
- 5) Clustering Coefficient – Współczynnik klastrowania, czyli tendencja do tworzenia lokalnych grup
- 6) Average Betweenness – Średnia centralność pośredniczenia, czyli jak często wierzchołki występują na najkrótszych ścieżkach
- 7) Average Closeness – Średnia centralność bliskości, czyli jak blisko są pozostałe wierzchołki od danego węzła
- 8) Average Eigenvector – Średnia centralność wektorowa, czyli ocena ważności połączeń wierzchołka
- 9) Largest CC Size – Rozmiar największej składowej spójności, czyli liczba wierzchołków w największej spójnej części grafu
- 10) Density – gęstość grafu, czyli stosunek liczby krawędzi do maksymalnej możliwej liczby krawędzi (czyli sytuacji, gdy każdy wierzchołek jest połączony z każdym innym)

## 2. Budowa środowiska

### Tworzenie grafów

```
import networkx as nx

# Main config
num_nodes = 100
num_edges = 2 # Edges for each new node
rewiring_prob = 0.1 # Probability of rewiring

# 1) Erdos-Renyi
prob_edge = 0.05 # Connection probability
erg_graph = nx.erdos_renyi_graph(num_nodes, prob_edge)

# 2) Barabasi-Albert
ba_graph = nx.barabasi_albert_graph(num_nodes, num_edges)

# 3) Watts-Strogatz
ws_graph = nx.watts_strogatz_graph(num_nodes, num_edges * 2,
rewiring_prob)

# 4) Stochastic Block Model (SBM)
sizes = [30, 30, 40] # Sizes of individual communities
p_intra = 0.6 # High probability of connections within communities
p_inter = 0.05 # Low probability of connections between communities
p_matrix = [[p_intra if i == j else p_inter for j in
range(len(sizes))]
for i in range(len(sizes))]
sbm_graph = nx.stochastic_block_model(sizes, p_matrix)

# 6) Planted Partition
num_communities = 4 # Number of communities
community_size = 25 # Size of each community
p_in = 0.5 # High probability of connection within communities
p_out = 0.05 # Low probability of connection between communities
planted_graph = nx.planted_partition_graph(num_communities,
community_size, p_in, p_out)

# 7) Karate Club
```

```

karate_graph = nx.karate_club_graph()

# 8) Random Geometric
radius = 0.3 # Radius within which connection occurs
rg_graph = nx.random_geometric_graph(num_nodes, radius)

# 9) Powerlaw Cluster
powerlaw_graph = nx.powerlaw_cluster_graph(num_nodes, num_edges, 0.1)

graphs_dict = {
    "Erdos-Renyi": erg_graph,
    "Barabasi-Albert": ba_graph,
    "Watts-Strogatz": ws_graph,
    "Stochastic Block Model (SBM)": sbm_graph,
    "Planted Partition": planted_graph,
    "Karate Club": karate_graph,
    "Random Geometric": rg_graph,
    "Powerlaw Cluster": powerlaw_graph
}

```

## Narzędzie do analizy grafu

```

import networkx as nx
import numpy as np

def analyze_graph(graph: nx.Graph, name: str) -> dict[str, float]:
    """Run specified graph analysis."""

    print(f'Analyzing graph {name}\t\t\t\t\t{graph}')

    degrees = [d for _, d in graph.degree()]
    avg_degree = np.mean(degrees)
    avg_shortest_path = nx.average_shortest_path_length(graph) if
nx.is_connected(graph) else None
    clustering_coeff = nx.average_clustering(graph)
    betweenness = nx.betweenness_centrality(graph)
    closeness = nx.closeness_centrality(graph)
    eigenvector = nx.eigenvector_centrality(graph, max_iter=1000,

```

```

tol=1e-6)

    largest_cc = max(nx.connected_components(graph), key=len) if
nx.is_connected(graph) else None
    largest_cc_size = len(largest_cc) if largest_cc else None
    density = nx.density(graph)

    return {
        "Graph Type": name,
        "Nodes": graph.number_of_nodes(),
        "Edges": graph.number_of_edges(),
        "Avg Degree": avg_degree,
        "Avg Shortest Path": avg_shortest_path,
        "Clustering Coefficient": clustering_coeff,
        "Avg Betweenness": np.mean(list(betweenness.values())),
        "Avg Closeness": np.mean(list(closeness.values())),
        "Avg Eigenvector": np.mean(list(eigenvector.values())),
        "Largest CC Size": largest_cc_size,
        "Density": density
    }

```

## Wywoływanie analizy dla utworzonych grafów

```

import matplotlib
import matplotlib.pyplot as plt
import networkx as nx
import pandas as pd

from graph_analyzer import analyze_graph
from graphs import graphs_dict

matplotlib.use('TkAgg')

if __name__ == '__main__':

    # Gather analysis
    results = [
        analyze_graph(graph, name=graph_name)
        for graph_name, graph in graphs_dict.items()
    ]

```

```

print()

# Create analysis dataframe
df_results = pd.DataFrame(results)
print(df_results.to_string())

# Plot graphs
graphs_per_row = round(len(graphs_dict) / 2)
fig, axes = plt.subplots(2, graphs_per_row, figsize=(3 *
graphs_per_row, 6))
axes_flat = axes.flatten()

for ax, (title, graph) in zip(axes_flat, graphs_dict.items()):
    nx.draw_spring(graph, ax=ax, node_size=20, edge_color="gray",
alpha=0.7)
    ax.set_title(title)

plt.show()

```

### 3. Przeprowadzenie eksperymentów

#### Wywołanie skryptu

Analyzing graph Erdos-Renyi and 251 edges	Graph with 100 nodes
Analyzing graph Barabasi-Albert nodes and 196 edges	Graph with 100
Analyzing graph Watts-Strogatz nodes and 200 edges	Graph with 100
Analyzing graph Stochastic Block Model (SBM) Graph named 'stochastic_block_model' with 100 nodes and 1149 edges	
Analyzing graph Planted Partition named 'stochastic_block_model' with 100 nodes and 782 edges	Graph

Analyzing graph Karate Club Graph named  
 "Zachary's Karate Club" with 34 nodes and 78 edges

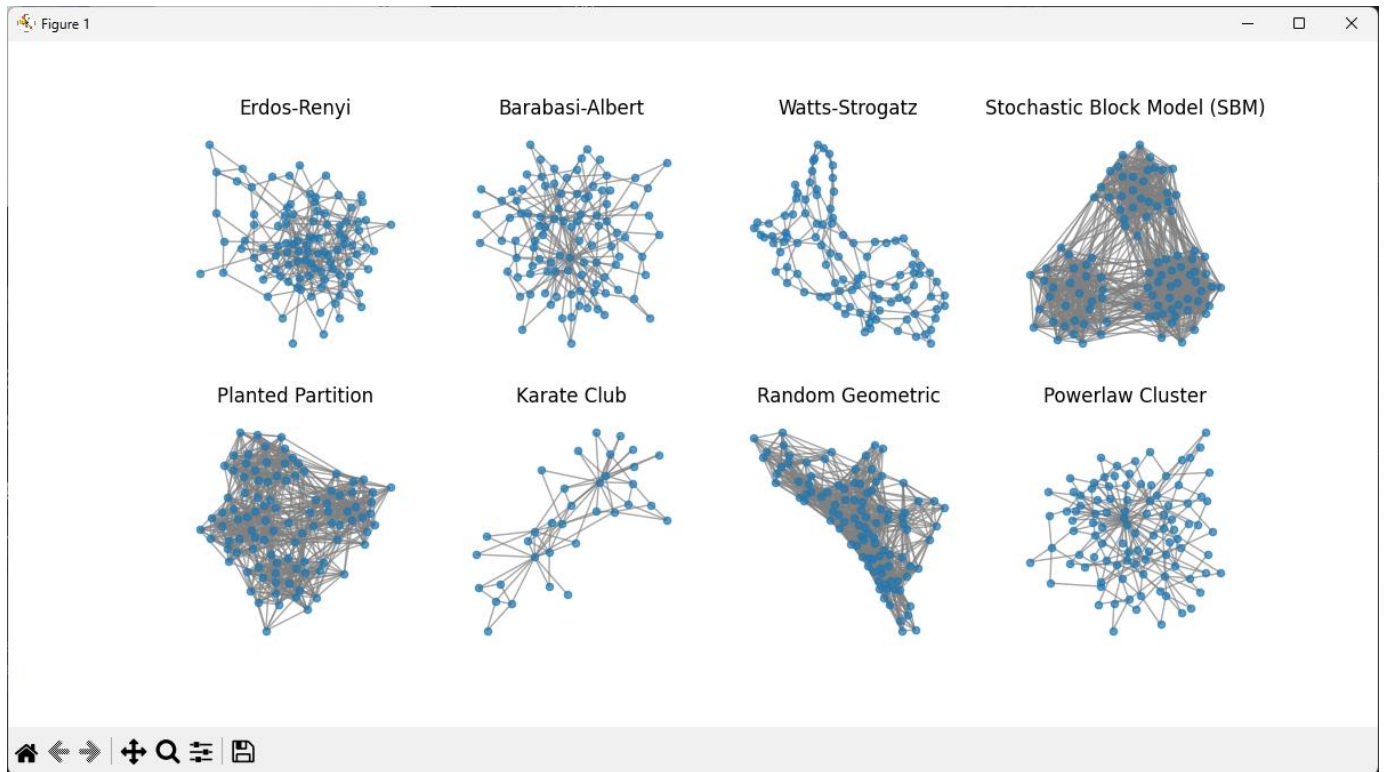
Analyzing graph Random Geometric Graph with 100  
 nodes and 1079 edges

Analyzing graph Powerlaw Cluster Graph with 100  
 nodes and 196 edges

	Graph Type	Nodes	Edges	Avg Degree	Avg
Shortest Path	Clustering Coefficient		Avg Betweenness	Avg Closeness	
Avg Eigenvector	Largest CC Size	Density			
0	Erdos-Renyi	100	251	5.020000	
3.052121	0.087157		0.020940	0.331834	
0.085486	100 0.050707				
1	Barabasi-Albert	100	196	3.920000	
2.991313	0.174359		0.020320	0.341167	
0.070703	100 0.039596				
2	Watts-Strogatz	100	200	4.000000	
5.156566	0.399667		0.042414	0.195671	
0.095045	100 0.040404				
3	Stochastic Block Model (SBM)	100	1149	22.980000	
1.849293	0.452996		0.008666	0.542051	
0.090339	100 0.232121				
4	Planted Partition	100	782	15.640000	
2.042020	0.330796		0.010633	0.491229	
0.097677	100 0.157980				
5	Karate Club	34	78	4.588235	
2.408200	0.570638		0.044006	0.426480	
0.146411	34 0.139037				
6	Random Geometric	100	1079	21.580000	
2.369899	0.686039		0.013979	0.433020	
0.083945	100 0.217980				
7	Powerlaw Cluster	100	196	3.920000	
2.927475	0.229835		0.019668	0.348222	
0.071857	100 0.039596				

Process finished with exit code 0

## Wizualizacja grafów



## 4. Opracowanie wyników i wyciąganie wniosków

Zestawienie wyników do tabeli

Graph Type	Nodes	Edges	Avg Degree	Avg Shortest Path	Clustering Coefficient	Avg Betweenness	Avg Closeness	Avg Eigenvector	Largest CC Size	Density
Erdos-Renyi	100	251	5.02	3.052121	0.087157	0.02094	0.331834	0.085486	100	0.050707
Barabasi-Albert	100	196	3.92	2.991313	0.174359	0.02032	0.341167	0.070703	100	0.039596



Watts-Strogatz	100	200	4	5.156566	0.399667	0.042414	0.195671	0.095045	100	0.040404
Stochastic Block Model (SBM)	100	1149	22.98	1.849293	0.452996	0.008666	0.542051	0.090339	100	0.232121
Planted Partition	100	782	15.64	2.04202	0.330796	0.010633	0.491229	0.097677	100	0.15798
Karate Club	34	78	4.588235	2.4082	0.570638	0.044006	0.42648	0.146411	34	0.139037
Random Geometric	100	1079	21.58	2.369899	0.686039	0.013979	0.43302	0.083945	100	0.21798
Powerlaw Cluster	100	196	3.92	2.927475	0.229835	0.019668	0.348222	0.071857	100	0.039596

## Analiza wartości

1. Nodes – Zadeklarowana taka sama liczba z wyłączeniem Karate Club (liczba 34 wierzchołków wynika z definicji grafu)
2. Edges – Najmniej Karate Club ze względu na małą liczbę wierzchołków, najwięcej dla SBM (ze względu na duże wartości prawdopodobieństw) oraz Random Geometric (ze względu na duży promień)
3. Average Degree – Największe wartości wynikające z liczby krawędzi, pozostałe wartości umiarkowane, co wskazuje na równomierne rozmieszczenie połączeń
4. Average Shortest Path – Wartości zbliżone z wyjątkiem Watts-Strogatz ze względu na niewielką liczbę połączeń między strukturami od siebie oddalonymi
5. Clustering Coefficient – Duża klasteryzacja dla Random Geometric (ze względu na łączenie się przy odpowiednio bliskim położeniu) oraz Karate Club (wynika z definicji – tworzą się 2 grupy)
6. Average Betweenness – Wyróżniająca się wartość dla Watts-Strogatz przez to, że są długie najkrótsze ścieżki prowadzące przez wiele wierzchołków (widać to dobrze na wizualizacji grafu)
7. Average Closeness – Największa wartość dla SBM, wynika to z bardzo dużej liczby różnych połączeń
8. Average Eigenvector – Największa dla Karate Club ze względu na małą liczbę węzłów i dużą bliskość wierzchołków wewnątrz grup
9. Largest CC Size – Wszystkie grafy są spójne dla wszystkich wierzchołków
10. Density – Brak wyróżniającej się dużej gęstości,

## 5. Propozycja kierunków dalszych badań

1. Zastosowanie kolejnych algorytmów – ze względu na wielkość dziedziny jakim są grafy istnieje niepoliczalnie duża liczba algorytmów, które można wykorzystać to tego oraz podobnych ćwiczeń

2. Wielokrotne generowanie grafów przez algorytmy z elementem losowości – pozwoliłoby to na sprawdzenie zależności między właściwościami grafu a czynnikami losowymi przy ich generowaniu
3. Zastosowanie większej liczby cech do analizy – wykorzystanie kolejnych danych pozwalających na analizę grafów
4. Testowanie odporności grafu na utratę losowych lub kluczowych wierzchołków – dokonywanie analiz grafów jak i utworzonych na ich podstawie kopii, które zostałyby pozbawionych albo losowych, albo istotnych wierzchołków. Istotność wierzchołka mogłaby zostać oceniona na podstawie jego stopnia, obecności na najkrótszych ścieżkach lub szybkością dostępu do innych punktów