

При завданні графа-зародку необхідно враховувати, що його матриця суміжності має бути симетричною.

Бібліотечна функція CNet створює мережевий графік, використовуючи функцію gplot із круговими координатами. Він дозволяє просту, але інтуїтивно зрозумілу візуалізацію цієї мережі (рисунок 3.3).

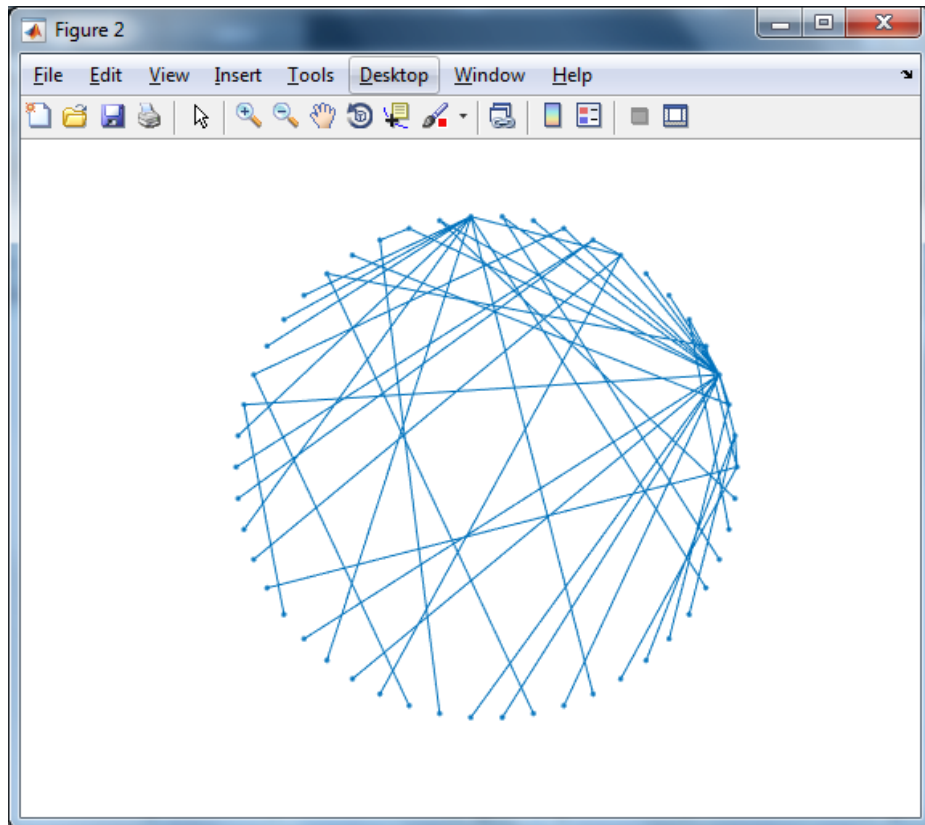


Рисунок 3.3 – Приклад візуалізації мережі

Функція PLplot приймає безмасштабну мережу у форматі матриці суміжності та виводить найкращу лінію, що апроксимує частотний розподіл степенів вузлів (рисунок 3.4). Функція повертає отримане рівняння апроксимованого степеневого розподілу.

Далі наведено приклад виклику описаних функцій (лістинг 3.1) та приклад залежності, яка апроксимує частотний розподіл степенів вузлів степеневою залежністю (рисунок 3.5). Коефіцієнт b є показником скейлінгу мінус один.

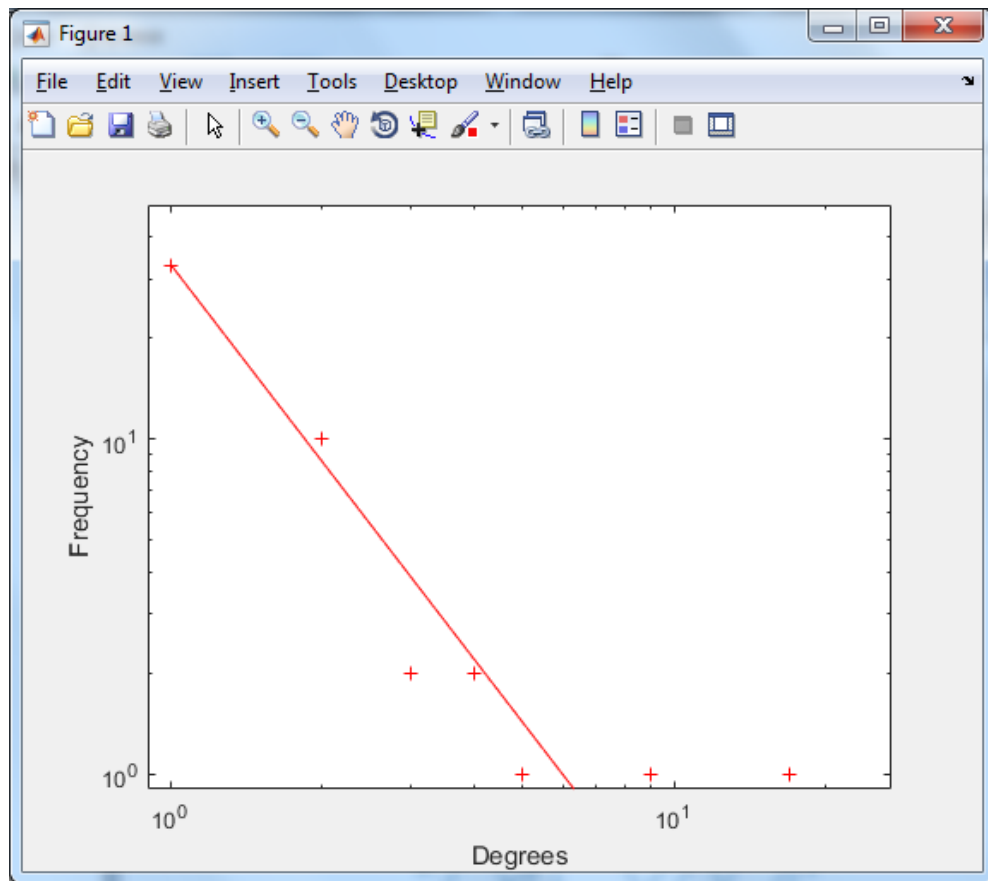


Рисунок 3.4 – Графік частотного розподілу степенів вузлів мережі

```

Command Window

New to MATLAB? See resources for Getting Started.

seed =

    0     1     0     0     1
    1     0     0     1     0
    0     0     0     1     0
    0     1     1     0     0
    1     0     0     0     0

PL_Equation =

General model Power1:
PL_Equation(x) = a*x^b
Coefficients (with 95% confidence bounds):
    a =    33.14   (30.13, 36.15)
    b =   -1.956  (-2.334, -1.578)

fx >>

```

Рисунок 3.5 – Приклад результатів роботи програми

У лістингу 3.1 наведено приклад коду виклику функції. У додатку А наведено тексти програм.

Лістинг 3.1 – Приклад коду виклику функцій

```
seed = [0 1 0 0 1; 1 0 0 1 0; 0 0 0 1 0; 0 1 1 0 0; 1 0 0 0 0]
Net = SFNG(50, 1, seed);
PL_Equation = PLplot(Net)
CNet(Net)
```

3.3 Програмна реалізація та моделювання еластичної безмасштабної мережі

Було проведено чисельне моделювання еластичної безмасштабної мережі з керуючими параметрами $\lambda = 1.3$, $p^{new} = 0.6$. Розмір мережі сягав $n = 65536$ вузлів. На рисунку 3.6 показано залежність степенів вузлів від їх порядкового номеру.

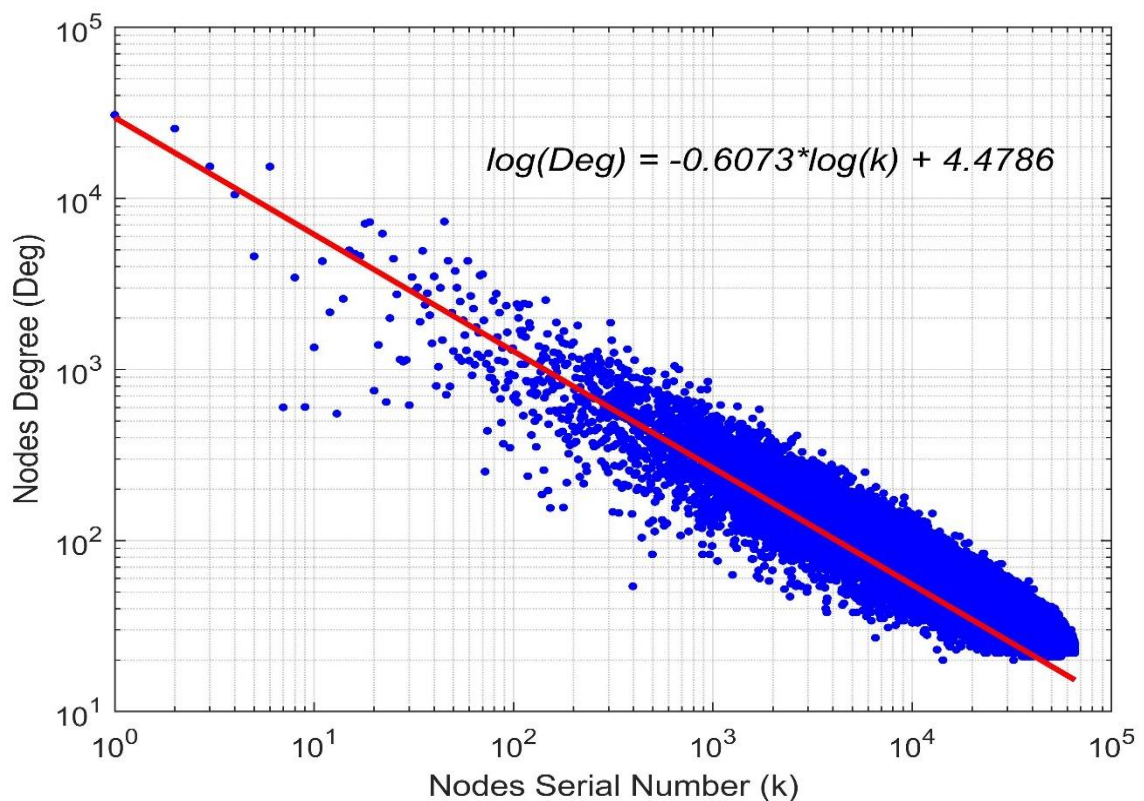


Рисунок 3.6 – Залежність степені вузлів від їх порядкового номеру

Можна бачити, що отриманий розподіл тісно збігається з теоретичним розподілом (0.54), який для даного прикладу повинен був мати вигляд:

$$deg_i = 16148 * i^{-0.61}. \quad (3.1)$$

Отже, можна сказати, що показник рангового скейлінгу (0.6073), отриманий експериментально, несуттєво відрізняється від теоретичного значення цього показника (0.61). Наведеним значенням рангового скейлінгу відповідають значення звичайного (частотного) скейлінгу $\gamma^{exper} = 2.647$ та $\gamma^{theor} = 2.639$. Отже, можна зробити висновок, що запропонована модель цілком придатна для моделювання безмасштабних мереж, забезпечуючи при цьому роздільну та незалежну параметризацію основних характеристик мережі – показника скейлінгу та показника еластичності.

ВИСНОВКИ

Розглянуто проблему моделювання безмасштабних мереж. Показано, що створення генеративних моделей таких мереж є актуальною проблемою. Вказано, що застосування неединичного показника еластичності кількості ребер відносно кількості вузлів мережі дає змогу створювати моделі щільних мереж у широкому діапазоні їх розміру. Зазначено, що для мереж реального світу показник скейлінгу розподілу вузлів за зв'язками варіюється. Зазвичай, в діапазоні від 2 до 3, є різним для різних мереж та не залежить від щільності чи еластичності. Тому актуальним завданням була така модифікація моделі еластичних мереж, в якій є можливість встановлювати найважливіші структурні характеристики мережі – скейлінг та еластичність – незалежно одну від одної. Для вирішення поставленого завдання було застосовано ймовірність під'єднання нового ребра до нового вузла. У традиційних моделях зазначена ймовірність є одиничною, проте у роботі було доведено, що варіація зазначеного параметра дає змогу генерувати безмасштабну мережу із довільним показником скейлінгу (20), який не залежить від заданої еластичності.

Проведено чисельний експеримент з моделювання безмасштабної мережі середнього розміру (65536 вузлів) із заданими показниками скейлінгу та еластичності. Результати моделювання підтверджують коректність запропонованої моделі та дотримання необхідного значення скейлінгу. Розроблена модель може бути застосована як для генерації безмасштабних мереж, так і з метою дослідження структурних властивостей існуючих мереж.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Albert R., Barabási A.-L. Statistical mechanics of complex networks. *Reviews of modern physics*. 2002. Vol. 74, no. 1. P. 47–97. URL: <https://doi.org/10.1103/revmodphys.74.47> (date of access: 05.06.2026).
2. Lietz H. Watts, Duncan J./Strogatz, Steven H. Collective dynamics of » small- world « networks. *Schlüsselwerke der Netzwerkforschung*. Wiesbaden, 2018. P. 551–553. URL: https://doi.org/10.1007/978-3-658-21742-6_130 (date of access: 05.06.2026).
3. Newman M. Power laws, Pareto distributions and Zipf's law. *Contemporary physics*. 2005. Vol. 46, no. 5. P. 323–351. URL: <https://doi.org/10.1080/00107510500052444> (date of access: 05.07.2026).
4. Dorogovtsev S. N., Mendes J. F. F., Samukhin A. N. Structure of growing networks with preferential linking. *Physical review letters*. 2000. Vol. 85, no. 21. P. 4633–4636. URL: <https://doi.org/10.1103/physrevlett.85.4633> (date of access: 05.07.2026).
5. Software for complex networks. *NetworkX*. URL: <https://networkx.org/documentation/stable/index.html> (date of access: 05.07.2026).
6. Elastic scale-free networks model based on the mediaton-driven attachment rule / V. Shergin et al. 2020 *IEEE third international conference on data stream mining & processing (DSMP)*, Lviv, Ukraine, 21–25 August 2020. 2020. URL: <https://doi.org/10.1109/dsmp47368.2020.9204207> (date of access: 05.07.2026).
7. Identifying time dependence in network growth / M. Falkenberg et al. *Physical review research*. 2020. Vol. 2, no. 2. URL: <https://doi.org/10.1103/physrevresearch.2.023352> (date of access: 05.07.2026).

8. Meng X., Zhou B. Scale-free networks beyond power-law degree distribution. *Chaos, solitons & fractals*. 2023. Vol. 176. P. 114173. URL: <https://doi.org/10.1016/j.chaos.2023.114173> (date of access: 05.07.2026).

9. Shergin V., Chala L. The concept of elasticity of scale-free networks. *2017 4th international scientific-practical conference problems of infocommunications. science and technology (PIC S&T)*, Kharkov, 10–13 October 2017. URL: <https://doi.org/10.1109/infocommst.2017.8246392> (date of access: 05.07.2026).

10. Unravelling the size distribution of social groups with information theory in complex networks / A. Hernando et al. *The european physical journal B*. 2010. Vol. 76, no. 1. P. 87–97. URL: <https://doi.org/10.1140/epjb/e2010-00216-1> (date of access: 05.07.2026).

11. Лазарєв Ю. Початки програмування у середовищі MatLAB : навч. посіб. Київ : Корнійчук, 1999. 160 с.

12. Герасімчук М., Лазарєв Ю., Толочко Т. Моделювання систем у середовищі MATLAB-SIMULINK: навч. посібник. Київ : НТУУ «КПІ», 2006. 175 с.

ДОДАТОК А

Тексти програм

```

function SFNet = SFNG(Nodes, mlinks, seed)

seed = full(seed);
pos = length(seed);

%if (Nodes < pos) || (mlinks > pos) || (pos < 1) ||
(size(size(seed)) ~= 2) || (mlinks < 1) || (seed ~= seed') ||
(sum(diag(seed)) ~= 0)
%   error('invalid parameter value(s)');
%end

%if mlinks > 5 || Nodes > 15000 || pos > 15000
%   warning('Abnormally large value(s) may cause long
processing time');
%end

rand('state',sum(100*clock));

Net = zeros(Nodes, Nodes, 'single');
Net(1:pos,1:pos) = seed;
sumlinks = sum(sum(Net));

while pos < Nodes
    pos = pos + 1;
    linkage = 0;
    while linkage ~= mlinks
        rnode = ceil(rand * pos);
        deg = sum(Net(:,rnode)) * 2;
        rlink = rand * 1;
        if rlink < deg / sumlinks && Net(pos,rnode) ~= 1 &&
Net(rnode,pos) ~= 1
            Net(pos,rnode) = 1;
            Net(rnode,pos) = 1;
            linkage = linkage + 1;
            sumlinks = sumlinks + 2;
        end
    end
end

clear Nodes deg linkage pos rlink rnode sumlinks mlinks
SFNet = Net;

function equation = PLplot(Net)
% Power-Law Degree Distribution Graphing

% Finds out how many connections each node has
connections = single(sum(Net));

```

```

% Initialize variable that will hold how many nodes have each
degree
frequency = single(zeros(1,length(Net)));

% Initialize variable that will hold the graphing quantities
plotvariables = zeros(2,length(Net));
P = [];

for T = 1:length(Net)
    % Variable will be used as a list of possible degrees a
    node can have
    P(1,T) = T;
    if connections(1,T) ~= 0
        frequency(1,connections(1,T)) =
frequency(1,connections(1,T)) + 1;
    end
end

for c = 1:length(frequency)
    % Disregard degrees with no frequency
    if frequency(1,c) ~= 0
        [X,Y] = find(plotvariables == 0);
        plotvariables(1,min(Y)) = P(1,c);
        plotvariables(2,min(Y)) = frequency(1,c);
    end
end

% Find the last non-zero element in plotvariables
for d = 1:length(plotvariables)
    if plotvariables(1,d) == 0 & plotvariables(2,d) == 0
        break
    end
end

x = plotvariables(1,1:d-1);
y = plotvariables(2,1:d-1);
[g,f,b] = fit(x',y','power1');
H = loglog(x,y,'r+');
hold on;
plot(g);
xlim([.9 (max(sum(Net)) + 10)]);
ylim([.9 length(Net)]);
legend off;
H = xlabel('Degrees');
H = ylabel('Frequency');
% Use this feature to extract variables from cfit variables
%a = g.a;
%b = g.b;
%rsquare = f.rsquare;
equation = g;

```

```

function CNet(Net)

format compact
format long e
theta = linspace(0,2*pi,length(Net)+1);
xy = zeros(length(Net)+1,2);
x = cos(theta);
y = sin(theta);
xy(1:length(Net)+1,1) = x(1:length(Net)+1);
xy(1:length(Net)+1,2) = y(1:length(Net)+1);
figure, gplot(Net,xy,'.-');
set(gcf, 'Color', [1 1 1]);
axis('equal');
xlim([-1.1 1.1]);
ylim([-1.1 1.1]);
axis off;

function dist = findDist(Net)
[n,i0] = max(sum(Net));
n = length(Net);
dist = -ones(n,1);
dist(i0) = 0;
R = 0;
nodes = setdiff(1:n,i0);
while (length(nodes)>0)
    ind = find(dist == R);
    for i=ind'
        j = intersect(find(Net(i,:)), nodes);
        dist(j) = R + 1;
        nodes = setdiff(nodes, j);
    end;
    R = R+1;
end;

function [S, dim] = fracDim(Net)
dist = findDist(Net);
n = max(dist);
for R = 1:n
    S(R) = sum(dist<=R);
end;
figure;
loglog(S,'-ob','LineWidth',2); grid on;
n = length(S);
dim = log(S')\[log(1:n)', ones(n,1)];

seed =[0 1 0 0 1;1 0 0 1 0;0 0 0 1 0;0 1 1 0 0;1 0 0 0 0];
Net = SFNG(5000, 1, seed);
S = fracDim(Net);
hold on;
k = 6;
dim = [log(1:k)', ones(k,1)]\log(S(1:k)');
loglog(exp(dim(2))*(1:k).^dim(1),'--r','LineWidth',2);
grid on; hold off;

```

ДОДАТОК Б

Відомість кваліфікаційної роботи

[illegible]