Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и

информатики» (СибГУТИ)

Отчет по **Лабораторной работе №5**

по дисциплине «Теория массового обслуживания»

Тема: «Марковские цепи. Исследование эргодических свойств»

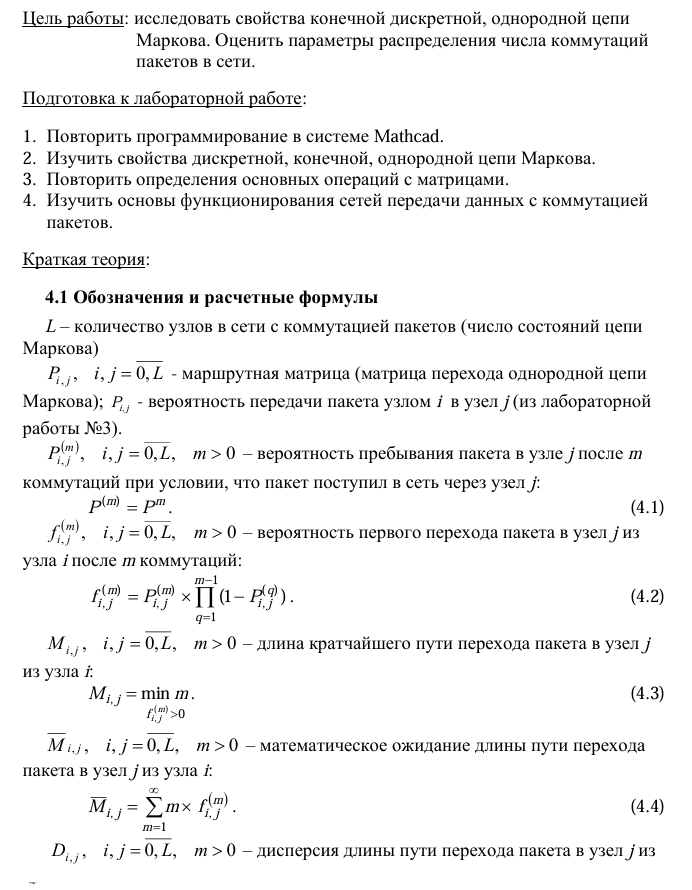
**Вариант 1**

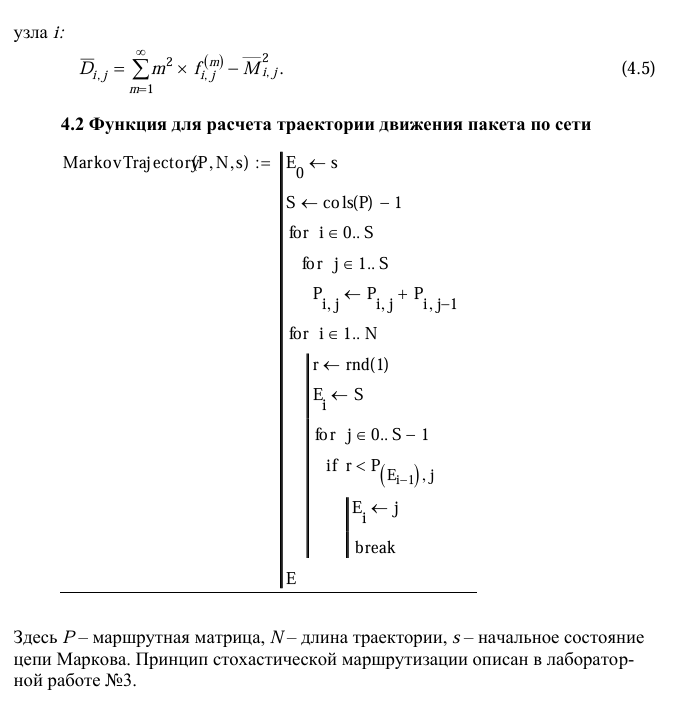
Выполнил:

студент гр. ИА-232

Московских Дмитрий Петрович

Новосибирск 2024





1. **Реализация функции MarkovTrajectory**

function E = MarkovTrajectory(P, N, s)

E = zeros(1, N);

E(1) = s;

S = size(P,1);

for i = 1:S

for j = 2:S

P(i, j) = P(i, j) + P(i, j - 1);

end

end

for i = 2:N

r = rand();

E(i) = S;

for j = 1:S

if r < P(E(i-1), j)

E(i) = j;

break;

end

end

end

end

Рассчёт траекторий:

trajectory = MarkovTrajectory(P,N,s);

figure;

plot(1:N, trajectory, '-o', 'LineWidth', 2);

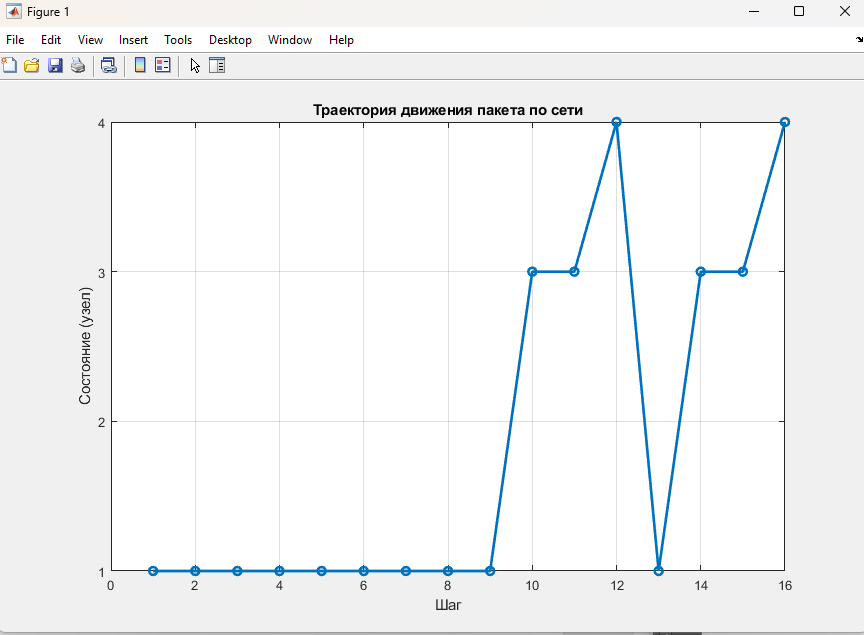
title('Траектория движения пакета по сети');

xlabel('Шаг');

ylabel('Состояние (узел)');

grid on;

yticks(1:size(P, 1));



1. Расчёт величин

\*вероятность пребывания пакета в узле j после m коммутаций при условии, что пакет поступил в сеть через узел i,

▪ вероятность первого перехода пакета в узел j из узла i после m коммутаций,

▪ длину кратчайшего пути перехода пакета в узел j из узла i,

▪ математическое ожидание длины пути перехода пакета в узел j из узла i,

▪ дисперсию длины пути перехода пакета в узел j из узла i,

Заданная точность:

epsilon = 1e-3;

Реализация функций:

% Функция для вычисления математического ожидания длины пути

function exp\_length = expected\_path\_length(P, i, j, epsilon)

m = 1;

exp\_length = 0;

while true

prob = first\_passage\_probability(P, i, j, m);

exp\_length = exp\_length + m \* prob;

if prob <= epsilon

break;

end

m = m + 1;

end

end

function prob = first\_passage\_probability(P, i, j, m)

if m == 1

prob = P(i, j);

else

prob = P(i, j) \* (1 - P(i, j))^(m-1);

end

end

% Функция для нахождения длины кратчайшего пути

function length = shortest\_path\_length(P, i, j)

length = inf;

for m = 1:100

if first\_passage\_probability(P, i, j, m) > 0

length = m;

break;

end

end

end

% Функция для вычисления дисперсии длины пути

function var\_length = variance\_path\_length(P, i, j, epsilon)

m = 1;

exp\_length = expected\_path\_length(P, i, j, epsilon);

var\_length = 0;

while true

prob = first\_passage\_probability(P, i, j, m);

var\_length = var\_length + m^2 \* prob;

if prob <= epsilon

break;

end

m = m + 1;

end

var\_length = var\_length - exp\_length^2;

end

% Вычисления

m = N; % Количество шагов

probs = zeros(L, L);

shortest\_paths = zeros(L, L);

expected\_lengths = zeros(L, L);

variances = zeros(L, L);

probs\_prebi = 0;

for i = 1:L

for j = 1:L

probs(i, j) = first\_passage\_probability(P, i, j, m);

shortest\_paths(i, j) = shortest\_path\_length(P, i, j);

expected\_lengths(i, j) = expected\_path\_length(P, i, j, epsilon);

variances(i, j) = variance\_path\_length(P, i, j, epsilon);

probs\_prebi = P.^i;

end

end

i = 1; j = 3; % Начальное и конечное состояния

disp(['Вероятность первого перехода: ', num2str(first\_passage\_probability(P, i, j, N))]);

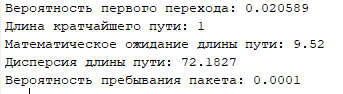
disp(['Длина кратчайшего пути: ', num2str(shortest\_path\_length(P, i, j))]);

disp(['Математическое ожидание длины пути: ', num2str(expected\_path\_length(P, i, j, epsilon))]);

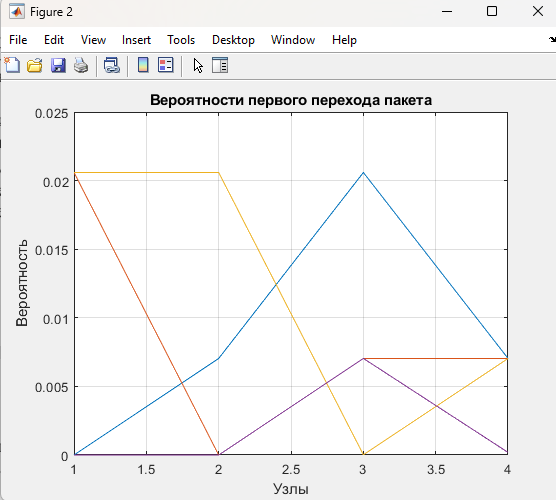
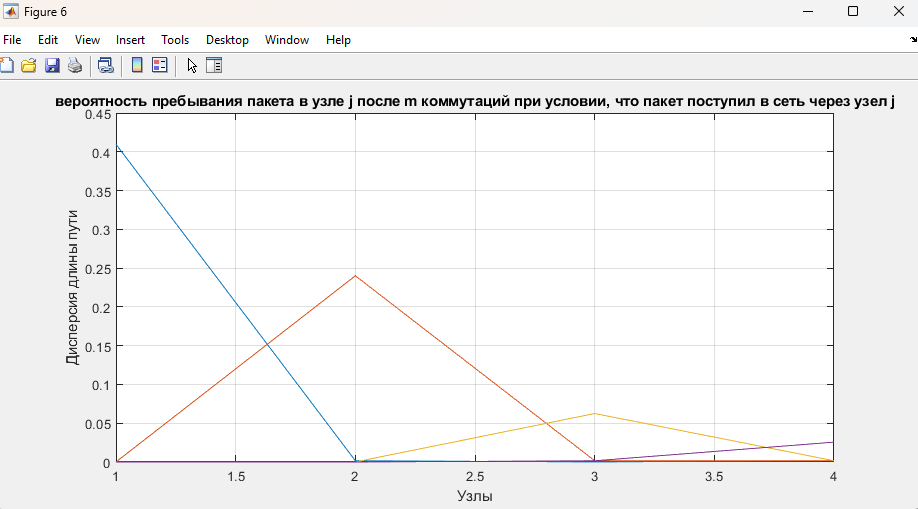
disp(['Дисперсия длины пути: ', num2str(variance\_path\_length(P, i, j, epsilon))]);

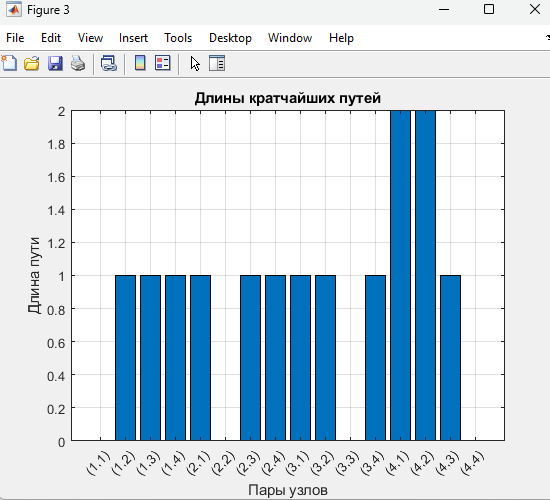
disp(['Вероятность пребывания пакета: ', num2str(probs\_prebi(i,j))]);

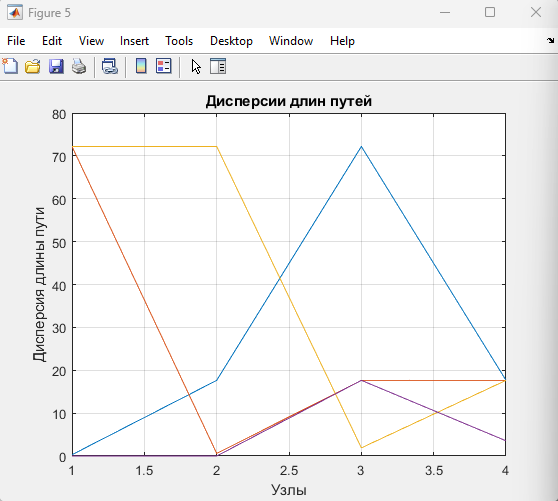
Результат

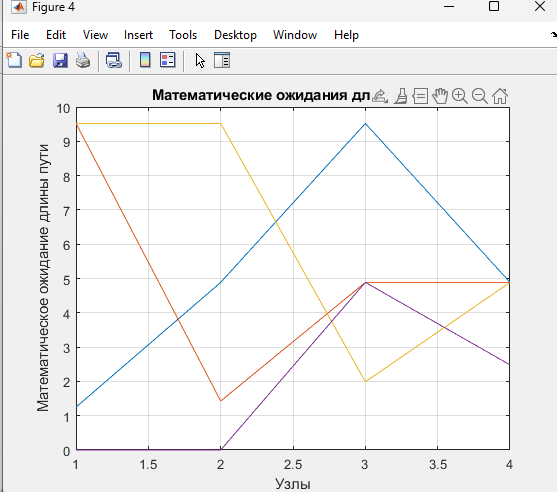


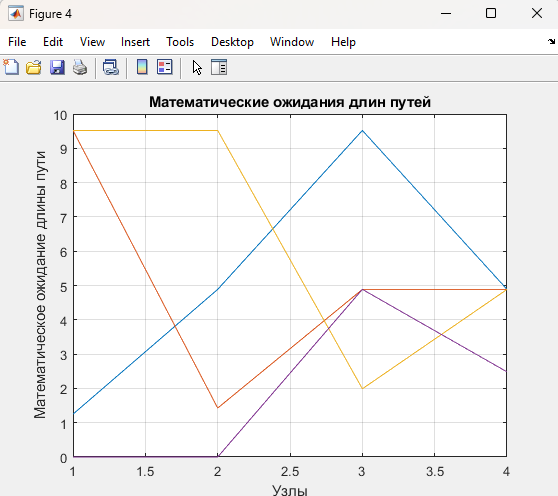
1. Зависимости

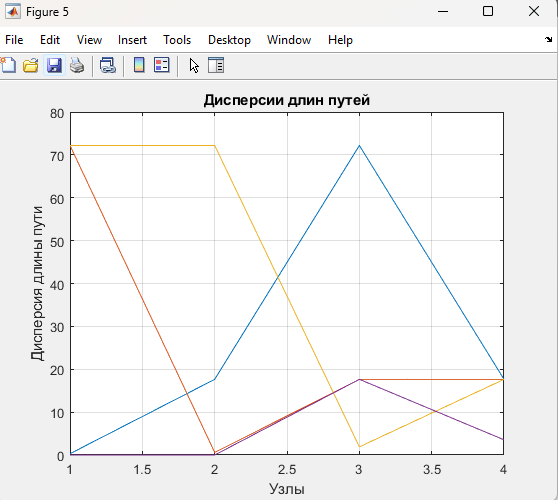












% Вероятности первого перехода пакета

figure;

plot(probs);

title('Вероятности первого перехода пакета');

xlabel('Узлы');

ylabel('Вероятность');

grid on;

% Создаем матрицу кратчайших путей (у вас уже есть это как shortest\_paths)

n = size(shortest\_paths, 1);

pairs = cell(n \* n, 1); % массив для пар узлов

index = 1;

% Формируем метки вида (i,j)

for i = 1:n

for j = 1:n

pairs{index} = sprintf('(%d,%d)', i, j);

index = index + 1;

end

end

% Построение графика

figure;

bar(shortest\_paths(:)); % столбчатая диаграмма

title('Длины кратчайших путей');

xlabel('Пары узлов');

ylabel('Длина пути');

set(gca, 'XTick', 1:(n\*n), 'XTickLabel', pairs); % установка меток оси X

xtickangle(45); % поворот меток оси X для удобства чтения

grid on;

% Математические ожидания длин путей

figure;

plot(expected\_lengths);

title('Математические ожидания длин путей');

xlabel('Узлы');

ylabel('Математическое ожидание длины пути');

grid on;

% Дисперсии длин путей

figure;

plot(variances);

title('Дисперсии длин путей');

xlabel('Узлы');

ylabel('Дисперсия длины пути');

grid on;

% Дисперсии длин путей

figure;

plot(probs\_prebi);

title('вероятность пребывания пакета в узле j после m коммутаций при условии, что пакет поступил в сеть через узел j');

xlabel('Узлы');

ylabel('Дисперсия длины пути');

grid on;

Контрольные вопросы:

**Определение цепи Маркова**

Цепь Маркова — это стохастический процесс, в котором вероятность перехода в следующее состояние зависит только от текущего состояния, а не от предыдущих состояний.

**Классификация цепей Маркова**

Цепи Маркова можно классифицировать по следующим признакам:

* **Дискретные и непрерывные:** по виду времени (дискретное или непрерывное).
* **Конечные и бесконечные:** по количеству состояний.
* **Однородные и неоднородные:** по зависимости переходных вероятностей от времени.

**Свойства цепей Маркова**

* **Марковское свойство:** будущее состояние зависит только от текущего состояния.
* **Стационарность:** свойства цепи не изменяются со временем.
* **Эргодичность:** в долгосрочной перспективе система посещает все состояния.

**Состояния цепи Маркова**

Состояния могут быть:

* **Поглощающие:** состояние, из которого невозможно выйти.
* **Переходные:** состояния, из которых возможен переход в другие состояния.

**Дискретные и непрерывные цепи Маркова**

* **Дискретные цепи Маркова:** переходы происходят в дискретные моменты времени.
* **Непрерывные цепи Маркова:** переходы происходят в непрерывное время.

**Вероятности перехода за m шагов**

Вероятности перехода за m шагов можно вычислить, возводя матрицу переходных вероятностей в степень m.

**Длины кратчайших путей перехода пакета**

Кратчайшие пути можно найти с помощью алгоритмов, таких как алгоритм Дейкстры или алгоритм Беллмана-Форда, которые минимизируют суммарное расстояние или стоимость переходов.

**Алгоритм функции вычисления траектории цепи Маркова**

Алгоритм вычисления траектории может включать:

1. Определение начального состояния.
2. Вычисление вероятностей перехода.
3. Генерация случайного числа для определения следующего состояния.
4. Повторение шагов до достижения конечного состояния.

**Общие понятия о вычислительных сетях с коммутацией пакетов и методах маршрутизации в них**

В вычислительных сетях с коммутацией пакетов данные разбиваются на пакеты, которые отправляются независимо друг от друга. Методы маршрутизации включают статическую маршрутизацию, где маршруты фиксированы, и динамическую маршрутизацию, где маршруты изменяются в зависимости от текущих условий сети.

**Вывод**

В ходе рассмотрения цепей Маркова и их применений было выявлено, что эти модели позволяют эффективно анализировать и прогнозировать поведение систем с вероятностными переходами. Такие знания особенно полезны при анализе вычислительных сетей и разработке эффективных алгоритмов маршрутизации, что обеспечивает надежную и оптимальную передачу данных в сетях.