МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
МИРЭА – РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

А.Б. СОРОКИН, Л.М. ЖЕЛЕЗНЯК

ПРОЕКТИРОВАНИЕ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ. МАРКОВСКИЕ СЛУЧАЙНЫЕ ПРОЦЕССЫ

ПРАКТИКУМ

Москва — 2023

УДК 519.852.3
ББК 22.18
С 65

Сорокин А.Б. Проектирование систем поддержки принятия решений. Марковские случайные процессы. [Электронный ресурс]: Практикум / Сорокин А.Б., Железняк Л.М. — М.: МИРЭА – Российский технологический университет, 2023. — 1 электрон. опт. диск (CD-ROM)

Практикум содержит задачи процессов Маркова и подходы к их решению. Рассматриваются марковские случайные процессы с дискретным временем и конечным числом состояний, а также непрерывные марковские случайные процессы с конечным числом состояний. Предназначено для студентов 4-го курса квалификации бакалавр, обучающихся по направлению 09.03.04 «Программная инженерия» и поддерживает дисциплину «Разработка обеспечивающих подсистем систем поддержки принятия решений» для проведения практических занятий.

Практикум издается в авторской редакции.

Авторский коллектив: Сорокин Алексей Борисович, Железняк Лилия Михайловна

Рецензенты:

Андреева Ольга Николаевна д.т.н., доцент, начальник отдела научной работы АО "Концерн "Моринсис-Агат"

Бражникова Елена Владимировна к.т.н., доцент кафедры вычислительной техники

Минимальные системные требования:
Наличие операционной системы Windows, поддерживаемой производителем.
Наличие свободного места в оперативной памяти не менее 128 Мб.
Наличие свободного места в памяти хранения (на жестком диске) не менее 30 Мб.
Наличие интерфейса ввода информации.
Дополнительные программные средства: программа для чтения pdf-файлов (Adobe Reader).
Подписано к использованию по решению Редакционно-издательского совета
МИРЭА – Российского технологического университета от ___ __________ 2023 г.
Объем ___ Мб
Тираж 10

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 4
1. МАРКОВСКИЕ СЛУЧАЙНЫЕ ПРОЦЕССЫ С ДИСКРЕТНЫМИ СОСТОЯНИЯМИ И ДИСКРЕТНЫМ ВРЕМЕНЕМ 5
1.1. Теоретические сведения 5
1.2. Решения практических задач 8
1.3. Программная реализация 14
2. СЛУЧАЙНЫЙ МАРКОВСКИЙ ПРОЦЕСС С ДИСКРЕТНЫМИ СОСТОЯНИЯМИ И НЕПРЕРЫВНЫМ ВРЕМЕНЕМ 19
2.1. Теоретические сведения 19
2.2. Решения практических задач 21
2.3. Программная реализация 24
ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ 29
ВАРИАНТЫ ДЛЯ ПРАКТИЧЕСКИХ РАБОТ 30
Задание 1. Марковские цепи с дискретным временем 30
Задание 1. Марковские цепи с непрерывным временем 37
СПИСОК ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 43
Сведения об авторах 44

ВВЕДЕНИЕ

Марковский процесс представляет собой определенную разновидность случайного процесса, то, есть случайной функции времени. Поэтому вначале необходимо определить, понятие случайной функции.

Случайным процессом называется изменение во времени состояния какой-либо системы в соответствии с вероятностными характеристиками. Обозначается как $X(t)$, где параметр t играет роль времени, а случайная величина X имеет некоторое распределение (дискретное и непрерывное). Случайный процесс характеризуется математическим ожиданием, дисперсией [1].

Таким образом случайный процесс , протекающий в любой физической среде S, представляет собой случайные переходы системы из одного состояния в другое. В зависимости от множества состояний и множества значений аргумента времени t все случайные процессы делятся на классы [2]:

1. Случайный процесс с дискретными состояниями и дискретным временем. В этом случае аргумент t принимает дискретные значения. Множество значений случайной величины X – конечное дискретное множество.

2. Случайный процесс с непрерывными состояниями и дискретным временем. В этом случае время t дискретно, случайная величина X имеет непрерывное распределение.

3. Случайный процесс с дискретными состояниями и непрерывным временем. В этом случае аргумент t непрерывно распределена на некотором интервале времени. Множество значений случайной величины X остается дискретным.

4. Случайный процесс с непрерывным состоянием и непрерывным временем.

Несмотря на то, что могут рассматриваться более сложные классы случайных процессов, упомянутые выше четыре являются основными.

Случайный процесс называется марковским, если для любого момента времени t_0 условная вероятность каждого из состояний системы S при $t > t_0$ зависит только от состояния при $t = t_0$ и не зависит только от состояний при $t < t_0$ [4]. Условие Маркова можно интерпретировать как: в любой момент времени прогноз о будущем поведении системы зависит от состояний, в которых находилась эта система в прошлом. Это условие Маркова позволяет найти простое математическое решение для ряда задач со случайными процессами.

1. МАРКОВСКИЕ СЛУЧАЙНЫЕ ПРОЦЕССЫ С ДИСКРЕТНЫМИ СОСТОЯНИЯМИ И ДИСКРЕТНЫМ ВРЕМЕНЕМ

1.1. Теоретические сведения

Пусть физическая система S находится в одно из состояний $S_1, S_2, \ldots, S_n$ и может переходить из одного состояния в другое случайным образом только в фиксированные изолированные моменты времени $t_1, t_2, \ldots, t_m$. И если на процесс наложено условие Маркова, то такой случайный процесс называется марковским случайным процессом с дискретным временем и конечным числом состояний (дискретными состояниями). Моменты времени еще называют шагами процесса, а сам марковский случайный процесс – марковской цепью [4].

Без ограничения общности рассмотрим случай S_n состояний. Обозначим через P_{ij} вероятности переходов системы из состояния S_i, в состояние S_j за один шаг. Такие вероятности образуют матрицу P_{ij} вероятностей переходов за один шаг. Поскольку система может пребывать в одном из n состояний, то для каждого момента времени t необходимо задать n^2 вероятностей перехода P_{ij}, которые удобно представить в виде следующей матрицы (1.1) [1 – 5]:

$$\|P_{ij}\| = \begin{pmatrix} P_{11} & P_{12} & \ldots & P_{1n} \\ P_{21} & P_{22} & \vdots & P_{2n} \\ \ldots & \ldots & & \ldots \\ P_{n1} & P_{n2} & \ldots & P_{nn} \end{pmatrix}. \tag{1.1}$$

Если вероятности P_{ij} не зависят от номера шага, на котором осуществляется переход $S_i \to S_j$, то такие цепи называются однородными марковскими цепями. В дальнейшем будем рассматривать только однородные марковские цепи.

Элементы матрицы обладают следующими свойствами [6]:

1. Значения вероятностей P_{ij} матрицы распределены от 0 до 1 (1.2).

$$0 \leq P_{ij} \leq 1, \ \ i, j = 1,2,3 \ldots n. \tag{1.2}$$

2. Строка матрицы характеризует одно из состояний системы S_i, при этом вероятности $P_i(k)$ определены шагом возможных переходов, также в строку входит переход P_{ii} сам на себя.

3. Столбец матрицы характеризует значения матрицы случайных переходов в выбранное состояние S_i системы. Таким образом строка матрицы – это переходы из состояния системы, а столбец – это переходы в состояния системы.

4. Сумма значений вероятностей матрицы каждой строки должна быть равна единице. Потому что, в строке определена полная сумма случайных событий (1.3).

$$\sum_{j=1}^{n} P_{ij} = 1, i = 1, 2, 3 \dots n. \tag{1.3}$$

5. В главной диагонали значения вероятностей определены как переходы P_{ii} сами на себя. Это обозначает что система не может выйти из этого состояния S_i, а всегда остается в нем.

Однородную марковскую цепь удобно изображать в виде размеченного графа состояний (рис. 1.1).

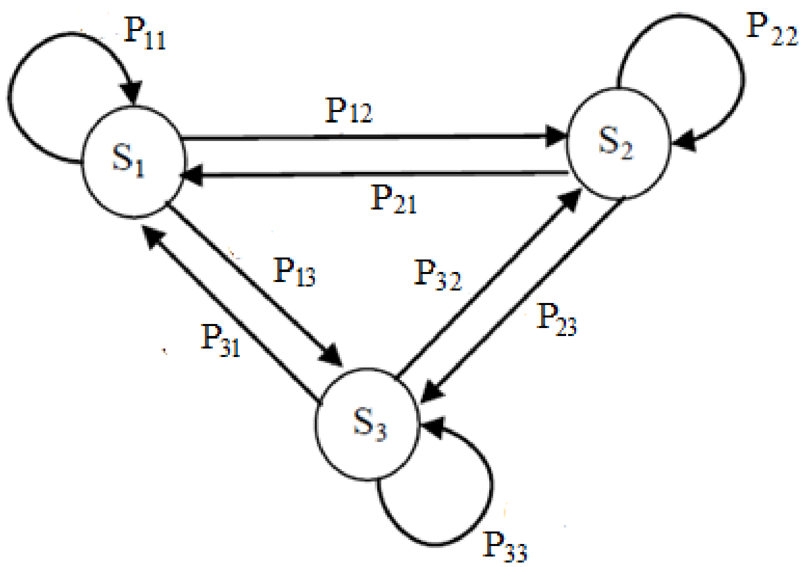

Рисунок 1.1. Граф состояний S_3

Обозначим через строку матрицы $Q(k) = (P_1(k), P_2(k), P_3(k))$ распределение вероятностей в системе после k шагов, где $P_i(k)$ – вероятность нахождения системы в состоянии P_i, , $i = 1, 2, 3$. Для элементов матрицы $Q(k)$ при любом $k \geq 0$ выполняется равенство $\sum_{i=1}^{3} P_i(k) = 1$. При $k = 0$ имеем начальное распределение вероятностей $Q(0) = (P_1(0), P_2(0), P_3(k))$ [7].

Для нахождения вероятностей после k-го шага справедливо следующее соотношение (1.4) [8]:

$$Q(k) = Q(k-1)P = Q(0)P^k \ . \tag{1.4}$$

Все состояния можно классифицировать на существенные и несущественные. Тогда состояние S_i называется существенным, если выйдя из этого состояния, система может в него вернуться за один или несколько шагов. Соответственно, состояние S_i называется несущественным, если выйдя из него, система не может вернуться в него [9].

Таким образом, марковская цепь, граф которой представлен на рис. 1.2. имеет одно несущественное состояние S_1 и два существенных S_2, S_3. Следовательно, если переходы (стрелки) только выходят из состояния, но не входят, то – это несущественное состояние.

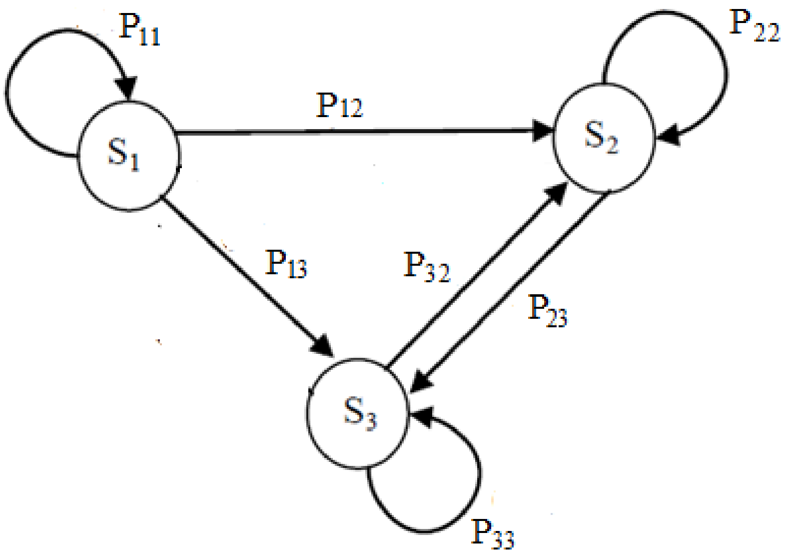

Рисунок 1.2. Граф состояний с несущественным состоянием S_1

Также можно цепь Маркова определить, как регулярную. Марковская цепь называется регулярной, если из любого существенного состояния можно попасть в любое другое существенное состояние за конечное число шагов.

Распределение вероятностей состояний можно классифицировать по времени. Распределение вероятностей состояний цепи Маркова называется стационарным, если оно не изменяется по времени. Стационарное распределение Q удовлетворяет матричному уравнению (1.5) [11].

$$Q = Q \times P. \tag{1.5}$$

Можно представить уравнение 1.5 в координатной форме (1.6, 1.7, 1.8)

$$p_1 = p_1 p_{11} + p_2 p_{21} + p_3 p_{31}. \tag{1.6}$$

$$p_2 = p_1p_{12} + p_2p_{22} + p_3p_{32}. \quad (1.7)$$

$$p_3 = p_1p_{13} + p_2p_{23} + p_3p_{33}. \quad (1.8)$$

Для получения единственного решения в системе уравнений необходимо добавить условие нормирования, которое определено свойством 4 элементов матрицы.

Элементы матрицы обладают следующими свойствами (1.9):

$$p_1 + p_2 + p_3 = 1. \quad (1.9)$$

Вероятности $\tilde{p}_i = \lim\limits_{k\to\infty} P_i(k)$ называют приделеньными или финальными вероятностями системы. Отсюда следует следующее утверждение: если марковская цепь регулярна, то предельные вероятности системы совпадают со стационарными вероятностями [12].

1.2. Решения практических задач

Рассмотрим решение марковских цепей.

Пример 1. Центральный процессор мультипрограммной системы в любой момент времени выполняет либо программы пользователя S_1, либо программы операционной системы S_2, либо находится в состоянии ожидания S_3. Граф состояний марковской модели имеет вид согласно рисунку 1.3. Длительности шага кратна продолжительности определения системы в каждом из состояний.

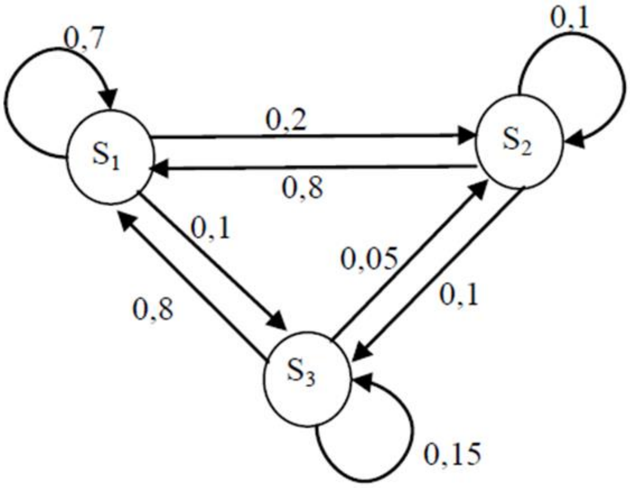

Рисунок 1.3. Размеченный граф вероятностных переходных состояний

Необходимо найти:

По размеченному марковскому графу найти матрицу вероятностей переходов.

Найти коэффициенты использования центрального процессора и затрачиваемое время на обработку программ и обслуживание системы.

Решение:

Составим матрицу вероятностных переходных процессов (1.10) согласно размеченному графу (рис. 1.3.) на основании свойств элементов матрицы.

$$\begin{pmatrix} 0,7 & 0,2 & 0,1 \\ 0,8 & 0,1 & 0,1 \\ 0,8 & 0,05 & 0,15 \end{pmatrix}. \qquad (1.10)$$

Исходя из матрицы переходных процессов определяется система линейных алгебраических уравнений (1.11).

$$\begin{cases} P_1 = 0,7P_1 + 0,8P_2 + 0,8P_3 \\ P_2 = 0,2P_1 + 0,1P_2 + 0,05P_3 \\ P_3 = 0,1P_1 + 0,1P_2 + 0,15P_3 \\ P_1 + P_2 + P_3 = 1 \end{cases}. \qquad (1.11)$$

На основании 4-го уравнения составляем уравнения вероятности состояний (1.12, 1.13 и 1.14):

$$P_1 = 1 - P_2 - P_3. \qquad (1.12)$$

$$P_2 = 1 - P_1 - P_3. \qquad (1.13)$$

$$P_3 = 1 - P_1 - P_2. \qquad (1.14)$$

Уравняем левые и правые части уравнений (1.15, 1.16 и 1.17).

$$1 - P_2 - P_3 = 0,7P_1 + 0,8P_2 + 0,8P_3. \qquad (1.15)$$

$$1 - P_1 - P_3 = 0,2P_1 + 0,1P_2 + 0,05P_3. \qquad (1.16)$$

$$1 - P_1 - P_3 = 0,1P_1 + 0,1P_2 + 0,15P_3. \qquad (1.17)$$

Тогда получаем (1.18, 1.19 и 1.20).

$$0{,}7P_1 + 1{,}8P_2 + 1{,}8P_3 = 1 \quad (1.18)$$

$$1{,}2P_1 + 0{,}1P_2 + 1{,}05P_3 = 1 \quad (1.19)$$

$$1{,}1P_1 + 1{,}1P_2 + 0{,}15P_3 = 1 \quad (1.20)$$

Решая систему методом Крамера, находим определители (1.21, 1.22, 1.23 и 1.24):

$$\Delta = \begin{pmatrix} 0{,}7 & 1{,}8 & 1{,}8 \\ 1{,}2 & 0{,}1 & 1{,}05 \\ 1{,}1 & 1{,}1 & 0{,}15 \end{pmatrix} = 0{,}7 \times 0{,}1 \times 0{,}15 + 1{,}8 \times 0{,}15 \times 1{,}1 + 1{,}8 \times 1{,}2 \times 1{,}1 - 1{,}8 \times 0{,}1 \times 1{,}1 - 0{,}7 \times 1{,}05 \times 1{,}1 - 1{,}8 \times 0{,}15 \times 1{,}2 = 3{,}135. \quad (1.21)$$

$$\Delta_1 = \begin{pmatrix} 1 & 1{,}8 & 1{,}8 \\ 1 & 0{,}1 & 1{,}05 \\ 1 & 1{,}1 & 0{,}15 \end{pmatrix} = 1 \times 0{,}1 \times 0{,}15 + 1{,}8 \times 0{,}15 \times 1 + 1{,}8 \times 1 \times 1{,}1 - 1{,}8 \times 0{,}1 \times 1 - 1 \times 1{,}05 \times 1{,}1 - 1{,}8 \times 1 \times 1{,}2 = 2{,}28. \quad (1.22)$$

$$\Delta_2 = \begin{pmatrix} 0{,}7 & 1 & 1{,}8 \\ 1{,}2 & 1 & 1{,}05 \\ 1{,}1 & 1 & 0{,}15 \end{pmatrix} = 0{,}7 \times 1 \times 0{,}15 + 1 \times 0{,}15 \times 1{,}1 + 1{,}8 \times 1{,}2 \times 1 - 1{,}8 \times 1 \times 1{,}1 - 0{,}7 \times 1{,}05 \times 1 - 1 \times 0{,}15 \times 1{,}2 = 0{,}525. \quad (1.23)$$

$$\Delta_3 = \begin{pmatrix} 0{,}7 & 1{,}8 & 1 \\ 1{,}2 & 0{,}1 & 1 \\ 1{,}1 & 1{,}1 & 1 \end{pmatrix} = 0{,}7 \times 0{,}1 \times 1 + 1{,}8 \times 1 \times 1{,}1 + 1 \times 1{,}2 \times 1{,}1 - 1 \times 0{,}1 \times 1{,}1 - 0{,}7 \times 1 \times 1{,}1 - 1{,}8 \times 1 \times 1{,}2 = 0{,}33. \quad (1.24)$$

В результате решения получаем значение вероятностей состояния в установленном режиме (1.25, 1.26 и 1.27).

$$P_1 = \frac{\Delta_1}{\Delta} = \frac{2{,}28}{3{,}135} \approx 0{,}727. \quad (1.25)$$

$$P_2 = \frac{\Delta_2}{\Delta} = \frac{0{,}525}{3{,}135} \approx 0{,}168. \quad (1.26)$$

$$P_3 = \frac{\Delta_3}{\Delta} = \frac{0{,}33}{3{,}135} \approx 0{,}105. \quad (1.27)$$

Таким образом, коэффициент не работы процессора $K_{\text{простоя}} = P_3 = 0{,}105$. Тогда коэффициент работы $K_{\text{использования}} = 1 – K_{\text{простоя}} = 0{,}895$. Следовательно, на обработку пользовательских программ используется 81,2% времени, а на обслуживание операционной системы – 18,8%.

Пример 2. Для марковской цепи с тремя состояниями задана матрица вероятностей переходов за один шаг (1.28):

$$P = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 \end{pmatrix}. \quad (28)$$

Необходимо:

1. Составить размеченный граф состояний этой марковской цепи, определить, является ли цепь регулярной.

2. Найти матрицу $P(2)$ вероятностей переходов за два шага.

3. Определить распределение вероятностей состояний системы за один, два и три шага, считая начальным состояние S_1.

4. Найти стационарное распределение вероятностей состояний.

5. Найти финальные вероятности состояний системы.

Решение:

1. Составим граф состояний (рис. 1.4).

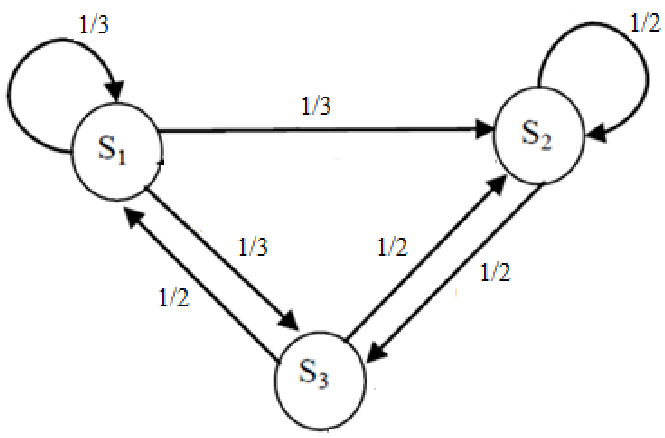

Рисунок 1.4. Размеченный граф состояний согласно заданной матрицы

По графу видно, что все состояния системы существенны и цепь регулярна.

2. Найдем матрицу вероятностей переходов за два шага (1.29).

$$P(2) = P \times P = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} \times \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = \begin{pmatrix} \frac{5}{18} & \frac{8}{18} & \frac{5}{18} \\ \frac{1}{4} & \frac{2}{4} & \frac{1}{4} \\ \frac{2}{12} & \frac{5}{12} & \frac{5}{12} \end{pmatrix}. \quad (1.29)$$

Заметим, что вероятность перехода из состояния S_2 в состояние S_1 за один шаг $p_{21}(1) = 0$, а за два шага $p_{21}(2) = \frac{1}{4} \neq 0$.

3. Пусть $Q = (1{,}0{,}0)$. Найдем распределение вероятностей состояний за один, два и три шага (1.30, 1.31).

$$Q(1) = Q(1) \times P = (1{,}0{,}0) \times \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = \left(\frac{1}{3}; \frac{1}{3}; \frac{1}{3}\right). \tag{1.30}$$

$$Q(2) = Q(0) \times P(2) = (1{,}0{,}0) \times \begin{pmatrix} \frac{5}{18} & \frac{8}{18} & \frac{5}{18} \\ \frac{1}{4} & \frac{2}{4} & \frac{1}{4} \\ \frac{2}{12} & \frac{5}{12} & \frac{5}{12} \end{pmatrix} = \left(\frac{5}{18}; \frac{8}{18}; \frac{5}{18}\right). \tag{1.31}$$

$Q(2)$ можно определить по-другому (1.32):

$$Q(2) = Q(1) \times P = \left(\frac{1}{3}; \frac{1}{3}; \frac{1}{3}\right) \times \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} =$$

$$= \left(\frac{5}{18}; \frac{8}{18}; \frac{5}{18}\right) = (0{,}28; 0{,}44; 0{,}28). \tag{1.32}$$

Таким образом, определим распределение вероятностей состояний после третьего шага (1.33).

$$Q(3) = Q(2) \times P = \left(\frac{5}{18}; \frac{8}{18}; \frac{5}{18}\right) \times \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} =$$

$$= \left(\frac{25}{108}; \frac{49}{108}; \frac{34}{108}\right) = (0{,}23; 0{,}45; 0{,}32). \tag{1.33}$$

4. Найдем стационарное распределение вероятностей (1.34, 1.35).

$$(p_1, p_2, p_3\,) \times \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = (p_1, p_2, p_3\,). \qquad (1.34)$$

$$\begin{cases} \frac{1}{3}p_1 + \frac{1}{2}p_3 = p_1 \\ \frac{1}{3}p_1 + \frac{1}{2}p_2 + \frac{1}{2}p_3 = p_2 \\ \frac{1}{3}p_1 + \frac{1}{2}p_2 = p_3 \end{cases}. \qquad (1.35)$$

Далее перенесем $(p_1, p_2, p_3\,)$ в левую часть уравнений, тем самым приравняем их к нулю (1.36).

$$\begin{cases} -\frac{2}{3}p_1 + \frac{1}{2}p_3 = 0 \\ \frac{1}{3}p_1 - \frac{1}{2}p_2 + \frac{1}{2}p_3 = 0 \\ \frac{1}{3}p_1 + \frac{1}{2}p_2 - p_3 = 0 \end{cases}. \qquad (1.36)$$

Система имеет бесчисленное множество решений, причем одно из уравнений является следствием двух других. Чтобы найти единственное решение, отбросим лишнее уравнение и добавим условие нормировки $p_1 + p_2 + p_3 = 1$.

Решим систему уравнений (1.37, 1.38, 1.39).

$$\begin{cases} -\frac{2}{3}p_1 + \frac{1}{2}p_3 = 0 \\ \frac{1}{3}p_1 - \frac{1}{2}p_2 + \frac{1}{2}p_3 = 0 \\ p_1 + p_2 + p_3 = 1 \end{cases} \cdot \qquad (1.37)$$

$$\begin{cases} -4p_1 + 3p_3 = 0 \\ 2p_1 - 3p_2 + 3p_3 = 0 \\ p_1 + p_2 + p_3 = 1 \end{cases}. \qquad (1.38)$$

$$Q_{\text{стац}} = \left(\frac{3}{13}; \frac{6}{13}; \frac{4}{13}\right) = (0{,}231; 0{,}461; 0{,}308). \qquad (1.39)$$

Марковская цепь регулярна, поэтому предельные вероятности совпадают со стационарными. Получим $\tilde{p}_1 = 3/13; \tilde{p}_2 = 6/13;\ \tilde{p}_3 = 4/13$. Можно заметить, что вероятности в первом и втором примере меняются на каждом шаге, но сумма $p_1 + p_2 + p_3 = 1$ остается неизменной. В разделе 1.3. приведен пример (листинга 1) на языке Питон и на языке Котлин в (листинге 2).

1.3. Программная реализация

Листинг 1. Программный код на языке Питон

```
import numpy as np
transition_matrix = np.array([[0.3, 0.2, 0, 0.5],
                              [0.1, 0.4, 0, 0.5],
                              [0.8, 0, 0.2, 0],
                              [0.4, 0.05, 0.5, 0.05]])
initial_state = np.array([float(x) for x in input("Введите начальное состояние вероятности: ").split(" ")])
time = int(input("Введите время: "))
print("Начальное состояние вероятности:", initial_state)
print(f"Требуется определить вероятности состояний через: {time}")
new_state = np.copy(initial_state)
for i in range(time):
    new_state = np.dot(new_state, transition_matrix)
    print(f"Вероятности состояний после {i} шага: \n{new_state}")

linear_equation = transition_matrix.T
print("Составим линейных алгебраических уравнений: ")
for idx1, row in enumerate(linear_equation):
    s = f"P{idx1 + 1} = "
    for idx2, ele in enumerate(row):
        s += f"{ele}P{idx2 + 1}"
        if idx2 != len(row) - 1:
            s += " + "
    print(s)
new_linear_equation = np.copy(linear_equation)
y = np.ones(new_linear_equation.shape[0])
for idx1, row in enumerate(new_linear_equation):
    s = ""
    for idx2, ele in enumerate(row):
        if idx1 == idx2:
            s += f"{ele}P{idx2 + 1}"
        else:
            new_linear_equation[idx1, idx2] += 1
            s += f"{new_linear_equation[idx1, idx2]}P{idx2 + 1}"
        if idx2 != len(row) - 1:
```

```
        s += " + "
    s += " = 1"
    print(s)

lambda0 = np.linalg.det([new_linear_equation])[0]
print("Лямбда =", lambda0)

lamb = []
for i in range(new_linear_equation.shape[1]):
    x = np.copy(new_linear_equation)
    x[:, i] = 1
    l = np.round(np.linalg.det([x])[0], 3)
    lamb.append(l)
    print(f"Лямбда{i+1} = {l}")

for i in range(len(lamb)):
    k = np.round(lamb[i] / lambda0, 3)
    print(f"Коэффициент K{i+1}: {k}")
```

Листинг 2. Программный код на языке Kotlin

```
import java.util.Vector
import kotlin.math.pow
import kotlin.collections.HashMap

//матрица вероятностей перехода
val matrix = HashMap<String, HashMap<String, Double>>()

//вектор начальных состояний
val vector = Vector<HashMap<String, Double>>()

fun main() {
buildMatrix()
buildVector()
println()
printMatrix()
println()
printFirstVector()
println()
```

```
makeTransition(7)
println()
findCoefficients()
}
fun buildMatrix() {
var line = HashMap<String, Double>()
line["Хорошее"] = 0.1
line["Обычное"] = 0.5
line["Плохое"] = 0.4
matrix["Хорошее"] = line

line = HashMap<String, Double>()
line["Хорошее"] = 0.3
line["Обычное"] = 0.4
line["Плохое"] = 0.3
matrix["Обычное"] = line

line = HashMap<String, Double>()
line["Хорошее"] = 0.2
line["Обычное"] = 0.5
line["Плохое"] = 0.3
matrix["Плохое"] = line

}
fun buildVector() {
val state = HashMap<String, Double>()
state["Хорошее"] = 1.0
state["Обычное"] = 0.0
state["Плохое"] = 0.0
vector.add(state)
}
fun printMatrix() {
print("Матрица вероятностей перехода настроений:\n\t")
for (key1 in matrix.keys)
print("\t$key1")
for (key1 in matrix.keys) {
print("\n$key1")
for (key2 in matrix[key1]!!.keys)
print("\t${String.format("%.4f", (matrix[key1]!![key2]))}")
}
println()
}
fun printFirstVector() {
println("Вектор начальных состояний P(0):")
for (key in vector.firstElement().keys) {
println("$key\t${String.format("%.4f", vector.firstElement()[key])}")
}
}
fun printLastVector() {
println("P(${vector.size - 1}):")
```

```
for (key in vector.lastElement().keys) {
println("$key\t${String.format("%.4f", vector.lastElement()[key])}")
}
}
fun makeTransition(num: Int) {
val state = HashMap<String, Double>()
for (key in matrix.keys) {
state[key] = 0.0
}
for (key1 in matrix.keys) {
for (key2 in matrix.keys) {
state[key2] = state[key2]!! + vector.lastElement()[key1]!! * matrix[key1]!![key2]!!
}
}
vector.add(state)
printLastVector()
if (num > 1) {
println()
makeTransition(num - 1)
}
}
fun findCoefficients() {
var array2 = arrayOf<Array<Double>>()
for (key in matrix.keys) {
var array1 = arrayOf<Double>()
for (key1 in matrix.keys)
for (key2 in matrix.keys)
if (key2 == key)
array1 += if (key1 == key2) matrix[key1]!![key2]!! else matrix[key1]!![key2]!! + 1
array2 += array1
}
println("Система уравнений:")
for (array in array2) {
var flag = true
for ((index, value) in array.withIndex())
if (flag) {
flag = false
print("(${String.format("%.4f", value)} * P${index + 1})")
} else print(" + (${String.format("%.4f", value)} * P${index + 1})")
println(" = 1")
}
val determinant = findDeterminant(array2)
println("\nDet = ${String.format("%.4f", determinant)}")
val determinants = Vector<Double>()
for (index in array2.indices) {
var newArray = arrayOf<Array<Double>>()
for (array1 in array2) {
var array = arrayOf<Double>()
array += array1.copyOfRange(0, index)
array += 1.0
```

Окончания листинга 2

```
array += array1.copyOfRange(index + 1, array1.size)
newArray += array
}
determinants.add(findDeterminant(newArray))
println("Det${index + 1} = ${String.format("%.4f", determinants.lastElement())}")
}
println()
for ((index, value)in determinants.withIndex()) {
println("P${index + 1} = ${String.format("%.4f", value / determinant)}")
}
println()
for ((index, key) in matrix.keys.withIndex()) {
println("Коэффициент \"$key\" составляет ${String.format("%.2f", determinants[index] /
determinant * 100)}%")
}
}
fun findDeterminant(array2: Array<Array<Double>>): Double {
if (array2.size == 1) return array2[0][0]
var determinant = 0.0
for ((rowIndex, array) in array2.withIndex()) {
var newArray = arrayOf<Array<Double>>()
for ((newRowIndex, array1) in array2.withIndex()) {
if (rowIndex != newRowIndex) {
newArray += array1.copyOfRange(1, array1.size)
}
}
determinant += array[0] * findDeterminant(newArray) * (-1.0).pow(2 + rowIndex)
}
return determinant
}
```

2. СЛУЧАЙНЫЙ МАРКОВСКИЙ ПРОЦЕСС С ДИСКРЕТНЫМИ СОСТОЯНИЯМИ И НЕПРЕРЫВНЫМ ВРЕМЕНЕМ

2.1. Теоретические сведения

Пусть физическая система S может находится в одном из состояний $S_1, S_2, \dots, S_n$ и будем считать, что переход из одного состояния в другое возможен в любой случайный момент времени t, причем множество таких моментов перехода является непрерывным. Если выполняется условие Маркова, то будем называть такой случайный процесс с конечным числом состояний (дискретными состояниями) и непрерывным временем или непрерывной марковской цепью. Рассмотрим, случай из трех состояний для упрощения выражений [13].

Интенсивностью перехода λ_{ij} перехода системы $S_i \to S_j$ называется предел (2.1).

$$\lambda_{ij} = \lim_{\Delta t \to 0} \frac{P_{ij(\Delta t)}}{\Delta t}. \tag{2.1}$$

где $P_{ij(\Delta t)}$ – вероятность перехода $S_i \to S_j$ $(i \neq j)$ на малом интервале времени $[t, t + \Delta\, t]$.

Величины λ_{ij} при $i \neq j$ в общем случае зависят от расположения интервала $[t, t + \Delta\, t]$ на оси времени (являются функциями времени). Величины $\lambda_{ij} = const$ при $i \neq j$, то марковская цепь называется однородной. Далее будем рассматривать только однородные марковские цепи [14].

Интенсивности λ_{ij} формируют матрицы интенсивности переходов (2.2).

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \end{pmatrix}. \tag{2.2}$$

Элементы матрицы интенсивностей Λ удовлетворяют условиям (2.3 – олд 2.5) [15]:

1. Значения плотностей вероятностей переходов матрицы λ_{ij} положительны (2.3).

$$\lambda_{ij} \geq 0 \text{ при } i \neq j \quad . \tag{2.3}$$

2. Значения диагоналей плотностей вероятностей переходов матрицы λ_{ii} отрицательны или равны нулю (2.4).

$$\lambda_{ii} \le 0 \text{ при } i = 1, 2, 3. \tag{2.4}$$

3. Сумма строки плотностей вероятностей переходов матрицы равны нулю (2.5).

$$\sum_{j=1}^{3} \lambda_{ij} = 0 \text{ при } i = 1, 2, 3. \tag{2.5}$$

Марковскую цепь с непрерывным временем можно изображать размеченным графом состояний (рис. 2.1).

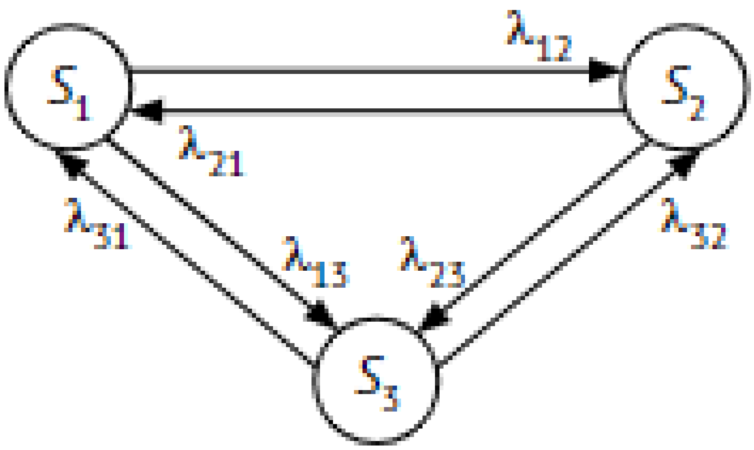

Рисунок 2.1. Марковскую цепь с непрерывным временем

Обозначим в виде $Q(t) = \big(p_1(t), p_2(t), p_3(t)\big)$ матрицу вероятностей состояний системы S в момент времени t, где $p_i(t)$ – вероятность того, что в момент времени t система находится в состоянии S_i. При любом t выполняется равенство $\sum_{j=1}^{3} p_i(t) = 1$. Для того чтобы найти эти вероятности необходимо решить систему дифференциальных уравнений Колмогорова (2.6) [10]:

$$\begin{cases} p_1' = \lambda_{11}p_1(t) + \lambda_{21}p_2(t) + \lambda_{31}p_3(t) \\ p_2' = \lambda_{12}p_1(t) + \lambda_{22}p_2(t) + \lambda_{32}p_3(t). \\ p_3' = \lambda_{13}p_1(t) + \lambda_{23}p_2(t) + \lambda_{33}p_3(t) \end{cases} \tag{2.6}$$

или в матричном виде (2.7):

$$Q'(t) = Q(t) \times \Lambda \ . \tag{2.7}$$

Для решения системы необходимо задать начальные условия (2.8).

$$p_1(0) = p_1^0, p_2(0) = p_2^0, p_3(0) = p_3^0 \ . \qquad (2.8)$$

где $p_i^0, (i = 1, 2, 3)$ – заданы неотрицательные значения, $p_1^0 + p_2^0 + p_3^0 = 1$.

Распределение вероятностей называется стационарным, если вероятности p_i состояния S_i не зависят от времени: $p_1(t) = p_1,\ p_2(t) = p_2\ ,\ p_3(t) = p_3$ или $Q(t) = Q = const$.

Для нахождения стационарных вероятностей необходимо решить систему алгебраических уравнений (2.9):

$$\begin{cases} \lambda_{11}p_1 + \lambda_{21}p_2 + \lambda_{31}p_3 = 0 \\ \lambda_{12}p_1 + \lambda_{22}p_2 + \lambda_{32}p_3 = 0. \\ \lambda_{13}p_1 + \lambda_{23}p_2 + \lambda_{33}p_3 = 0 \end{cases} \qquad (2.9)$$

или в матричном виде (2.10)

$$Q \times \Lambda = 0. \qquad (2.10)$$

Для многих практических задач важно знать, как ведут себя вероятности состояний системы при неограниченном возрастании времени. Введем предельные (финальные) вероятности системы $\tilde{p}_i = \lim\limits_{t \to \infty} p_i\ (t)$.

Понятия существенного (несущественного) состояния, регулярности марковской цепи, а также совпадение предельных и стационарных вероятностей в регулярной цепи аналогичны тем, что даны для марковской модели с дискретным временем. Система, для которой существуют предельные вероятности состояний, называется эргодической, а возникающий в ней в ней случайный процесс – эргодическим [15].

2.2. Решения практических задач

Пример. Задана матрица интенсивностей марковской цепи с непрерывным временем (2.11).

$$\Lambda = \begin{pmatrix} -2 & 1 & 1 \\ 3 & -4 & 1 \\ 0 & 1 & -1 \end{pmatrix}. \qquad (2.11)$$

Составить граф состояний, записать систему дифференциальных уравнений Колмогорова для вероятностных состояний. Найти частное решение системы при начальных условиях $p_1(0) = 1,\ p_2(0) = 0\ ,\ p_3(0) = 0$. Найти стационарное и предельное распределение вероятностей состояний системы.

Решение:

Граф данной марковской цепи изображен на рис. 2.2.

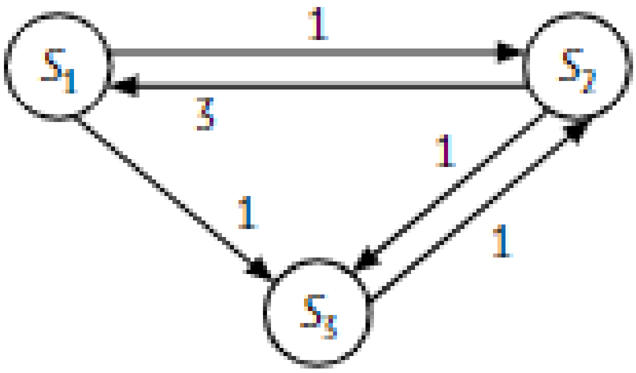

Рисунок 2.2. Граф данной марковской цепи

Составим систему дифференциальных уравнений Колмогорова (2.12).

$$(p_1'(t), p_2'(t), p_3'(t)) = \big(p_1(t), p_2(t), p_3(t)\big) \times \begin{pmatrix} -2 & 1 & 1 \\ 3 & -4 & 1 \\ 0 & 1 & -1 \end{pmatrix}. \quad (2.12)$$

Тогда получим (2.13).

$$\begin{cases} p_1'(t) = -2p_1(t) + 3\,p_2(t) \\ p_2'(t) = p_1(t) - 4\,p_2(t) + p_3(t). \\ p_3'(t) = p_1(t) + \ p_2(t) - p_3(t) \end{cases} \quad (2.13)$$

Найдем стационарные вероятности системы, пологая, что $p_1' = 0, p_2' = 0, p_3' = 0$.

Получим систему линейных алгебраических уравнений (2.14):

$$\begin{cases} -2p_1 + 3\,p_2 = 0 \\ p_1 - 4\,p_2 + p_3 = 0. \\ p_1 + \ p_2 - p_3 = 0 \end{cases} \quad (2.14)$$

Отбросим третье уравнение и присоединим к системе уравнений условие $p_1 + p_2 - p_3 = 1$ (2.15).

$$\begin{cases} -2p_1 + 3\,p_2 = 0 \\ p_1 - 4\,p_2 + p_3 = 0. \\ p_1 + p_2 + p_3 = 1 \end{cases} \qquad (2.15)$$

Находим решение: $p_1 = 3/10; p_2 = 2/10; p_3 = 5/10$, т.е. $Q = (0{,}3;\, 0{,}2;\, 0{,}5)$.

Для отыскания предельного распределения вероятностей состояний будем решать систему дифференциальных уравнений. Исключая переменную $p_3 = 1 - p_1 - p_2$, получим (2.16).

$$\begin{cases} p_1' = -2p_1 + 3\,p_2 \\ p_2' = -5\,p_2 + 1 \end{cases}. \qquad (2.16)$$

Сведем систему двух уравнений первого порядка к линейному дифференциальному уравнению второго порядка (2.17).

$$p_1'' + 7p_1' + 10p_1 = 3. \qquad (2.17)$$

Составим характеристическое уравнение, которое имеет вид (2.18):

$$k^2 + 7k + 10 = 0\ . \qquad (2.18)$$

с корнями $k_1 = -5$; $k_2 = -2$. Общее решение соответствующего однородного уравнения принимает следующий вид (2.19):

$$\bar{p}_1(t) = C_1 e^{-5t} + C_2 e^{-2t}. \qquad (2.19)$$

Частное решение будем искать в виде $p_1^* = A$. Подставляя в дифференциальное уравнение, получаем $A = 3/10$. Тогда общее решение неоднородного уравнения примет вид (2.20):

$$p_1(t) = C_1 e^{-5t} + C_2 e^{-2t} + 3/10. \qquad (2.20)$$

Находим $p_2(t)$ (2.21):

$$p_2(t) = \frac{1}{3}(p_1' + 2p_1) = -C_1 e^{-5t} + \frac{2}{10}. \tag{2.21}$$

Используя начальные условия $p_1(0) = 1, p_2(0) = 0$, запишем (2.22)

$$\begin{cases} 1 = C_1 + C_2 + \frac{3}{10} \\ 0 = -C_1 + \frac{2}{10} \end{cases}. \tag{2.22}$$

и определяем $C_1 = \frac{2}{10}, C_2 = \frac{5}{10}$. После этого находим $p_3(t)$ (2.23):

$$p_3(t) = 1 - p_1(t) - p_2(t) = \frac{5}{10} e^{-2t} + \frac{5}{10}\ . \tag{2.23}$$

Окончательно рассчитываем (2.24 – 2.26):

$$p_1(t) = \frac{2}{10} e^{-5t} + \frac{5}{10} e^{-2t} + \frac{3}{10}\ . \tag{2.24}$$

$$p_2(t) = -\frac{2}{10} e^{-5t} + \frac{2}{10}. \tag{2.25}$$

$$p_3(t) = -\frac{5}{10} e^{-2t} + \frac{5}{10}. \tag{2.26}$$

Эти решения описывают переходной процесс, протекающий в рассматриваемой марковской цепи. При $t \to \infty$ получим финальное распределение вероятностей (2.27)

$$\tilde{Q} = (\tilde{p}_1, \tilde{p}_2, \tilde{p}_3) = (0{,}3; 0{,}2; 0{,}5). \tag{2.27}$$

которое совпадает со стационарным распределением.

2.3. Программная реализация

В программное реализации представлен (листинг 3 и 4) на разных языках.

Листинг 3. Программный код на языке Питон

```
import numpy as np
density_matrix = np.array([[0, 2, 0, 5],
                           [1, 0, 0, 5],
                           [8, 0, 0, 0],
```

```
                    [4, 0.5, 5, 0]])
kolmogorov = []
n_state = 4
print("Система дифференциальных уравнений Колмогорова: ")
for i in range(n_state):
    m = []
    for j in range(i):
        m.append(density_matrix[j, i])
    r = np.sum(density_matrix[i]) * -1
    m.append(r)
    for j in range(i+1, n_state):
        m.append(density_matrix[j, i])
    kolmogorov.append(m)
kolmogorov = np.array(kolmogorov)
print(kolmogorov)
final_kolmogorov = np.copy(kolmogorov)
print("Финальные вероятности состояний: ")
final_kolmogorov[-1, :] = 1
print(final_kolmogorov)
lambd = np.linalg.det([final_kolmogorov])[0]
print("Лямбда =", np.round(lambd))
lamb = []
for i in range(final_kolmogorov.shape[1]):
    x = np.copy(final_kolmogorov)
    x[:-1, i] = 0
    l = np.round(np.linalg.det([x])[0], 3)
    lamb.append(l)
    print(f"Лямбда{i} = {l}")
print("Решение системы имеет вид:")
for i in range(len(lamb)):
    k = np.round(lamb[i] / lambd, 3)
    print(f"P{i}: {k}")
```

Листинг 4. Код программы маркоские непрерывные цепи

```
#include <iostream>
#include <iostream>
#include<vector>
#include<map>
#include<unordered_map>
#include<algorithm>
#include<string>
#include<math.h>

using namespace std;
/*
```

```
1

0 1 2 0

2 0 0 2
3 0 0 1
0 3 2 0
2

*/

//str for str
struct P {
   vector<double>values;
   P() {
      values.resize(4);
   }
};

double getDet(vector<vector<double>>& matrix) {
   double koef1 = matrix[0][0]; double koef2 = -1 * matrix[0][1]; double koef3 = matrix[0][2]; double koef4 =
-1 * matrix[0][3];
   double det1 = matrix[1][1] * matrix[2][2] * matrix[3][3] - matrix[1][1] * matrix[2][3] * matrix[3][2]
      - matrix[1][2] * matrix[2][1] * matrix[3][3] + matrix[1][2] * matrix[2][3] * matrix[3][1]
      + matrix[1][3] * matrix[2][1] * matrix[3][2] - matrix[1][3] * matrix[2][2] * matrix[3][1];

   double det2 = matrix[1][0] * matrix[2][2] * matrix[3][3] - matrix[1][0] * matrix[2][3] * matrix[3][2]
      - matrix[1][2] * matrix[2][0] * matrix[3][3] + matrix[1][2] * matrix[2][3] * matrix[3][0]
      + matrix[1][3] * matrix[2][0] * matrix[3][2] - matrix[1][3] * matrix[2][2] * matrix[3][0];

   double det3 = matrix[1][0] * matrix[2][1] * matrix[3][3] - matrix[1][0] * matrix[2][3] * matrix[3][1]
      - matrix[1][1] * matrix[2][0] * matrix[3][3] + matrix[1][1] * matrix[2][3] * matrix[3][0]
      + matrix[1][3] * matrix[2][0] * matrix[3][1] - matrix[1][3] * matrix[2][1] * matrix[3][0];

   double det4 = matrix[1][0] * matrix[2][1] * matrix[3][2] - matrix[1][0] * matrix[2][2] * matrix[3][1]
      - matrix[1][1] * matrix[2][0] * matrix[3][2] + matrix[1][1] * matrix[2][2] * matrix[3][0]
      + matrix[2][2] * matrix[2][0] * matrix[3][1] - matrix[2][2] * matrix[2][1] * matrix[3][0];

   return koef1 * det1 + koef2 * det2 + koef3 * det3 + koef4 * det4;
}

void init(vector<vector<double>>& a) {
   cout << "Введите значения для переходов в 4x4 матрице" << endl;
   for (int i = 0; i < 4;i++) {
      for (int j = 0; j < 4;j++) {
         cin >> a[i][j];
      }
   }
}

void solution(vector<vector<double>>& a) {
   P* str1 = new P();
```

```
    P* str2 = new P();
    P* str3 = new P();
    P* str4 = new P();
    str1->values[1] = a[1][0]; str1->values[2] = a[2][0]; str1->values[0] = -(a[0][1] + a[0][2]); str1->values[3] =
0;
    str2->values[0] = a[0][1]; str2->values[3] = a[3][1]; str2->values[1] = -(a[1][0] + a[1][3]); str2->values[2] =
0;
    str3->values[0] = a[0][2]; str3->values[3] = a[3][2]; str3->values[2] = -(a[2][0] + a[2][3]); str3->values[1] =
0;
    str4->values[1] = a[1][3]; str4->values[2] = a[2][3]; str4->values[3] = -(a[3][1] + a[3][2]); str4->values[0] =
0;
    //system with str1,str2,str3 and P0+P1+P2+P3 = 1
    str4->values[0] = 1; str4->values[1] = 1; str4->values[2] = 1; str4->values[3] = 1;
    vector<double>pi = { 0,0,0,1 };
    vector<vector<double>>matrix(4, vector<double>(4));
    matrix[0] = str1->values;
    matrix[1] = str2->values;
    matrix[2] = str3->values;
    matrix[3] = str4->values;
    double det = getDet(matrix);
    vector<vector<double>>matrix2 = matrix;
    matrix2[0][0] = pi[0]; matrix2[1][0] = pi[1]; matrix2[2][0] = pi[2]; matrix2[3][0] = pi[3];
    double m2 = getDet(matrix2);
    vector<vector<double>>matrix3 = matrix;
    matrix3[0][1] = pi[0]; matrix3[1][1] = pi[1]; matrix3[2][1] = pi[2]; matrix3[3][1] = pi[3];
    double m3 = getDet(matrix3);
    vector<vector<double>>matrix4 = matrix;
    matrix4[0][2] = pi[0]; matrix4[1][2] = pi[1]; matrix4[2][2] = pi[2]; matrix4[3][2] = pi[3];
    double m4 = getDet(matrix4);
    vector<vector<double>>matrix5 = matrix;
    matrix5[0][3] = pi[0]; matrix5[1][3] = pi[1]; matrix5[2][3] = pi[2]; matrix5[3][3] = pi[3];
    double m5 = getDet(matrix5);
    cout << m2 / det << ' ' << m3 / det << ' ' << m4 / det << ' ' << m5 / det << endl;
}

void menu(vector<vector<double>>&a) {
    int move;
    while (true) {
        cout << "Выберите действие:" << endl;
        cout << "1 - ввод матрицы" << endl;
        cout << "2 - вывод решения" << endl;
        cin >> move;
        switch (move) {
        case 0:
            return;
        case 1:
            init(a);
            break;
        case 2:
            solution(a);
            break;
        default:
```

Окончания листинга 4

```
            cout << "Попробуйте ещё раз" << endl;
        }
    }
}


int main()
{
    setlocale(LC_ALL, "Russian");
    vector<vector<double>>a(4, vector<double>(4));//4x4matrix
    menu(a);
}
```

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Что из себя представляет марковский процесс?
2. Что называется, случайным процессом?
3. На какие классы делятся случайные процессы?
4. Как можно охарактеризовать случайный процесс с дискретными состояниями и дискретным временем?
5. Как можно охарактеризовать случайный процесс с непрерывными состояниями и дискретным временем?
6. Как можно охарактеризовать случайный процесс с дискретными состояниями и непрерывным временем?
7. Как можно охарактеризовать случайный процесс с непрерывным состоянием и непрерывным временем?
8. Как можно интерпретировать условие Маркова?
9. Какими свойствами обладают элементы матрицы вероятностей переходов?
10. Чему равна сумма значений матрицы вероятностей переходов каждой строки?
11. Какие значения устанавливаются на главной диагонали матрицы вероятностей переходов?
12. Как изображать однородную марковскую цепь в виде размеченного графа состояний?
13. Какие состояния можно классифицировать на существенные?
14. Какая марковская цепь называется регулярной?
15. Какие состояния можно классифицировать по времени как стационарные?
16. Что называют придельными вероятностями системы?
17. Напишите формулу интенсивности перехода.
18. Чему равна интенсивность перехода в однородной марковской цепи?
19. Каким условиям удовлетворяют элементы матрицы интенсивностей?
20. Чему равна сумма значений матрицы интенсивностей каждой строки?
21. Какие значения принимают диагонали плотностей вероятностей переходов матрицы?
22. Приведите пример системы дифференциальных уравнений Колмогорова.
23. Какая система называется эргодической?

ВАРИАНТЫ ДЛЯ ПРАКТИЧЕСКИХ РАБОТ

Задание 1. Марковские цепи с дискретным временем

Пусть дана матрица переходных процессов за один шаг марковской цепи с дискретным временем. Составить граф марковской цепи. Реализовать программу для нахождения вероятности переходов из одного состояния в другое, распределение вероятностей на восемь шагов. Программная реализация также должна определять стационарные вероятности и сравнивать их за восемь шагов. Данные представлены в таблице 1 «Варианты заданий марковской цепи с дискретным временем».

Таблица 1. Варианты заданий марковской цепи с дискретным временем

Вариант	Задание
1	$P = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$
2	$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$
3	$P = \begin{pmatrix} \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$

Продолжение табл. 1

Вариант	Задание
4	$P = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$
5	$P = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{2}{3} & 0 & \frac{1}{3} \end{pmatrix}$
6	$P = \begin{pmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$
7	$P = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$

Продолжение табл. 1

Вариант	Задание
8	$P=\begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}$
9	$P=\begin{pmatrix} \frac{4}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$
10	$P=\begin{pmatrix} \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$
11	$P=\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{3}{4} & \frac{1}{4} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \end{pmatrix}$

Продолжение табл. 1

Вариант	Задание
12	$P=\begin{pmatrix}\frac{1}{6} & \frac{2}{3} & \frac{1}{6}\\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4}\\ \frac{1}{3} & \frac{2}{3} & 0\end{pmatrix}$
13	$P=\begin{pmatrix}\frac{1}{6} & \frac{2}{3} & \frac{1}{6}\\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6}\\ \frac{1}{4} & \frac{3}{4} & 0\end{pmatrix}$
14	$P=\begin{pmatrix}\frac{1}{4} & \frac{1}{4} & \frac{1}{2}\\ \frac{5}{6} & 0 & \frac{1}{6}\\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6}\end{pmatrix}$
15	$P=\begin{pmatrix}\frac{1}{6} & \frac{1}{6} & \frac{2}{3}\\ 0 & \frac{1}{2} & \frac{1}{2}\\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{pmatrix}$

Продолжение табл. 1

Вариант	Задание
16	$P=\begin{pmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2}\\ \frac{1}{6} & \frac{1}{3} & \frac{1}{2}\end{pmatrix}$
17	$P=\begin{pmatrix}\frac{1}{2} & \frac{1}{6} & \frac{1}{3}\\ \frac{1}{2} & 0 & \frac{1}{2}\\ \frac{3}{4} & \frac{1}{4} & 0\end{pmatrix}$
18	$P=\begin{pmatrix}\frac{1}{4} & \frac{1}{2} & \frac{1}{4}\\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3}\\ \frac{3}{4} & \frac{1}{4} & 0\end{pmatrix}$
19	$P=\begin{pmatrix}\frac{1}{4} & \frac{1}{4} & \frac{1}{2}\\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3}\\ \frac{3}{4} & \frac{1}{4} & 0\end{pmatrix}$

Продолжение табл. 1

Вариант	Задание
20	$P = \begin{pmatrix} \frac{1}{4} & 0 & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$
21	$P = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$
22	$P = \begin{pmatrix} \frac{3}{4} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$
23	$P = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$

Продолжение табл. 1

Вариант	Задание
24	$P = \begin{pmatrix} \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{3}{4} & \frac{1}{4} \end{pmatrix}$
25	$P = \begin{pmatrix} 0 & \frac{3}{4} & \frac{1}{4} \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$
26	$P = \begin{pmatrix} 0 & \frac{1}{3} & \frac{2}{3} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{5}{6} & \frac{1}{6} & 0 \end{pmatrix}$
27	$P = \begin{pmatrix} 0 & \frac{5}{6} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$

Окончание табл. 1

Вариант	Задание
28	$P=\begin{pmatrix}\frac{1}{9} & \frac{5}{9} & \frac{3}{9}\\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3}\\ \frac{2}{7} & \frac{3}{7} & \frac{2}{7}\end{pmatrix}$
29	$P=\begin{pmatrix}\frac{1}{4} & \frac{2}{4} & \frac{1}{4}\\ \frac{2}{10} & \frac{3}{10} & \frac{5}{10}\\ \frac{1}{2} & \frac{1}{2} & 0\end{pmatrix}$
30	$P=\begin{pmatrix}\frac{1}{11} & \frac{5}{11} & \frac{5}{11}\\ \frac{1}{5} & \frac{2}{5} & \frac{2}{5}\\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{pmatrix}$

Задание 2. Марковские цепи с непрерывным временем

Пусть задана матрица интенсивностей переходов непрерывной цепи Маркова. Составить размеченный граф состояний системы, соответствующий матрице, записать систему дифференциальных уравнений Колмогорова. Реализовать программу для нахождения частного решения системы при начальных условиях $p_1(0)=1, p_2(0)=0, p_3(0)=0$. Определить стационарные и предельные распределения вероятностей. Данные представлены в таблице 2 «Варианты заданий марковской цепи с непрерывным временем».

Таблица 2. Варианты заданий марковской цепи с непрерывным временем

Вариант	Задание
1	$\Lambda = \begin{pmatrix} -2 & 0 & 2 \\ 2 & -3 & 1 \\ 1 & 0 & -1 \end{pmatrix}$
2	$\Lambda = \begin{pmatrix} -4 & 2 & 2 \\ 0 & -3 & 3 \\ 1 & 1 & -2 \end{pmatrix}$
3	$\Lambda = \begin{pmatrix} -4 & 1 & 3 \\ 0 & -3 & 3 \\ 2 & 0 & -2 \end{pmatrix}$
4	$\Lambda = \begin{pmatrix} -6 & 1 & 5 \\ 3 & -4 & 1 \\ 2 & 0 & -2 \end{pmatrix}$
5	$\Lambda = \begin{pmatrix} -2 & 2 & 0 \\ 2 & -5 & 3 \\ 1 & 3 & -4 \end{pmatrix}$
6	$\Lambda = \begin{pmatrix} -3 & 1 & 2 \\ 1 & -3 & 2 \\ 1 & 1 & -2 \end{pmatrix}$
7	$\Lambda = \begin{pmatrix} -5 & 3 & 2 \\ 1 & -5 & 4 \\ 0 & 1 & -1 \end{pmatrix}$

Продолжение табл. 2

Вариант	Задание
8	$\Lambda = \begin{pmatrix} -4 & 2 & 2 \\ 1 & -2 & -1 \\ 1 & 1 & -2 \end{pmatrix}$
9	$\Lambda = \begin{pmatrix} -5 & 0 & 5 \\ 2 & -3 & 1 \\ 1 & 3 & -4 \end{pmatrix}$
10	$\Lambda = \begin{pmatrix} -7 & 2 & 5 \\ 1 & -5 & 4 \\ 0 & 3 & -3 \end{pmatrix}$
11	$\Lambda = \begin{pmatrix} -5 & 3 & 2 \\ 1 & -2 & 1 \\ 3 & 2 & -5 \end{pmatrix}$
12	$\Lambda = \begin{pmatrix} -8 & 5 & 3 \\ 2 & -5 & 3 \\ 1 & 7 & -8 \end{pmatrix}$
13	$\Lambda = \begin{pmatrix} -7 & 3 & 4 \\ 2 & -4 & 2 \\ 4 & 3 & -7 \end{pmatrix}$
14	$\Lambda = \begin{pmatrix} -3 & 2 & 1 \\ 2 & -2 & 0 \\ 2 & 5 & -7 \end{pmatrix}$

Продолжение табл. 2

Вариант	Задание
15	$A = \begin{pmatrix} -6 & 3 & 3 \\ 2 & -3 & 1 \\ 3 & 4 & -7 \end{pmatrix}$
16	$A = \begin{pmatrix} -6 & 2 & 4 \\ 3 & -3 & 0 \\ 1 & 5 & -6 \end{pmatrix}$
17	$A = \begin{pmatrix} -2 & 1 & 1 \\ 3 & -8 & 5 \\ 4 & 4 & -8 \end{pmatrix}$
18	$A = \begin{pmatrix} -4 & 2 & 2 \\ 3 & -9 & 6 \\ 3 & 5 & -8 \end{pmatrix}$
19	$A = \begin{pmatrix} -5 & 2 & 3 \\ 4 & -6 & 2 \\ 2 & 2 & -4 \end{pmatrix}$
20	$A = \begin{pmatrix} -7 & 2 & 5 \\ 1 & -2 & 1 \\ 3 & 1 & -4 \end{pmatrix}$
21	$A = \begin{pmatrix} -9 & 4 & 5 \\ 1 & -2 & 1 \\ 2 & 2 & -4 \end{pmatrix}$

Продолжение табл. 2

Вариант	Задание
22	$A = \begin{pmatrix} -3 & 2 & 1 \\ 4 & -6 & 2 \\ 3 & 3 & -6 \end{pmatrix}$
23	$A = \begin{pmatrix} -4 & 2 & 2 \\ 5 & -7 & 2 \\ 1 & 1 & -2 \end{pmatrix}$
24	$A = \begin{pmatrix} -5 & 3 & 2 \\ 7 & -8 & 1 \\ 1 & 1 & -2 \end{pmatrix}$
25	$A = \begin{pmatrix} -2 & 1 & 1 \\ 2 & -3 & 1 \\ 2 & 2 & -4 \end{pmatrix}$
26	$A = \begin{pmatrix} -4 & 3 & 1 \\ 4 & -8 & 4 \\ 2 & 3 & -5 \end{pmatrix}$
27	$A = \begin{pmatrix} -5 & 3 & 2 \\ 7 & -9 & 2 \\ 1 & 1 & -2 \end{pmatrix}$
28	$A = \begin{pmatrix} -6 & 5 & 1 \\ 2 & -4 & 2 \\ 3 & 3 & -6 \end{pmatrix}$

Окончание табл. 2.

Вариант	Задание
29	$\Lambda = \begin{pmatrix} -6 & 2 & 4 \\ 1 & -4 & 3 \\ 3 & 3 & -6 \end{pmatrix}$
30	$\Lambda = \begin{pmatrix} -4 & 2 & 2 \\ 7 & -10 & 3 \\ 4 & 4 & -7 \end{pmatrix}$

СПИСОК ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Башуров В.В. Марковские случайные процессы в моделировании систем: учебно-методическое пособие / В.В. Башуров, О.А. Башурова, А.П. Садов. – Екатеринбург: УрГУПС, 2020. – 100 с.

2. Дворяткина С. Н. Марковские процессы и простейшие модели теории массового обслуживания: учебное пособие / С.Н. Дворяткина, О.Н. Прокуратова – М.: «ФЛИНТА», 2022. – 80 с.

3. Лифшиц М.А. Случайные процессы — от теории к практике / М.А. Лифшиц – СПб.: Лань, 2022. – 308 с.

4. Свешников А.А. Прикладные методы теории марковских процессов / А.А. Свешников – СПб.: Лань, 2022. – 192 с.

5. Шихеева В.В. Теория случайных процессов. Марковские цепи / В.В. Шихеева – М.: «МИСИС», 2013. – 70 с.

6. Ганичева А.В. Теория вероятностей: учебное пособие / А.В. Ганичева – СПб.: Лань, 2022. – 144 с.

7. Симушкин С.В. Методы теории вероятностей: учебное пособие / С.В. Симушкин – СПб.: Лань, 2020. – 548 с.

8. Белякова А.Ю. Имитационное моделирование: Учебное пособие / А.Ю. Белякова – Молодежный: Изд-во ИрГАУ, 2020. – 120 с.

9. Катаргин Н.В. Анализ и моделирование логистических систем / Н.В. Катаргин, О.Н. Ларин, Ф.Д. Венде – СПб.: Лань, 2021. – 248 с.

10. Рыжиков Ю.И. Численные методы теории очередей: учебное пособие / Ю.И. Рыжиков – СПб.: Лань, 2022. – 512 с.

11. Певзнер Л.Д. Практикум по математическим основам теории систем / Л.Д. Певзнер – СПб.: Лань, 2022. – 400 с.

12. Катаргин Н.В. Сетевые модели в задачах экономики: учебник / Н.В. Катаргин, В.П. Невежин – СПб.: Лань, 2020. – 172 с.

13. Коршунов Д.А. Сборник задач и упражнений по теории вероятностей / Д.А. Коршунов, С.Г. Фосс, И.М. Эйсымонт– СПб.: Лань, 2022. – 220 с.

14. Свешников А.А. Сборник задач по теории вероятностей, математической статистике и теории случайных функций / А.А. Свешников – СПб.: Лань, 2022. – 448 с.

15. Белопольская Я.И. Стохастические дифференциальные уравнения. Приложения к задачам математической физики и финансовой математики: учебное пособие для вузов / Я.И. Белопольская – СПб.: Лань, 2021. – 308 с.

Сведения об авторах

Сорокин **Алексей Борисович**, кандидат технических наук, доцент, доцент кафедры вычислительной техники института Информационных технологий (РТУ МИРЭА);

Железняк **Лилия Михайловна**, старший преподаватель кафедры вычислительной техники института Информационных технологий (РТУ МИРЭА).