МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки, специальности)

Технология программирования
(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему «Реализация алгоритма управления марковской цепью»

Студент	В.В. Шило (И.О. Фамилия)	(личная подпись)
Руководитель	Г.А. Тырыгина (И.О. Фамилия)	(личная подпись)
Консультанты	К. А. Селиверстова (И.О. Фамилия)	(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент кафедры ПМИ, А.В. Очеповский
(ученая степень, звание, И.О. Фамилия)

(личная подпись)

«_____» _________________ 20___ г.

Тольятти 2019

АННОТАЦИЯ

Тема данной квалификационной работы: «Реализация алгоритма управления марковской цепью»

Актуальность бакалаврской работы обусловлена необходимостью создания программы, реализующей алгоритм управления марковской цепью.

Целью данной выпускной квалификационной работы является разработка программы, реализующей алгоритм управления марковской цепью. Разработанная в ходе работы программа может применяться для осуществления поиска наилучшей стратегии при формировании плана доходов и расходов.

Объектом исследования является марковская цепь.

Предмет исследования — марковская цепь с доходами.

Был осуществлён обзор методов управления марковской цепью и выбран алгоритм для реализации программы.

На основе объектно-ориентированного подхода разработана программа. В качестве среды разработки выбран язык программирования C++.

Выпускная квалификационная работа состоит из пояснительной записки объёмом в 40 стр. и содержит 4 рисунка, 4 таблицы и список литературы, который включает в себя 20 источников.

Работа состоит из введения, трёх глав, заключения и списка использованной литературы.

ABSTRACT

The title of the graduation work is «The implementation of the Markov chain control algorithm».

The key issue of the graduation work is due to the need of create a program that implements the Markov chain control algorithm.

The purpose of the graduation work is to develop a program that implements the Markov chain control algorithm. The program developed in this work can be aimed at finding the best strategy when forming a plan for income and expenses.

The object of this work is the Markov chain.

The subject of this work is a program that implements the Markov chain control algorithm.

Research methods: object-oriented programming.

A review of Markov chain control methods was implemented and an algorithm was chosen for program implementation.

As a result program was developed by an object-oriented approach. The programming language C++ was chosen as the development environment.

The graduation work consists of an explanatory note on 40 pages, including 4 figures, 4 tables and a list of references including 20 sources 5 foreign sources.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Марковский процесс — является частным случаем случайного процесса, у которого эволюция после какого-либо значения, которое было бы задано для параметра времени t не будет зависеть от эволюции, предшествовавшей t, при условии, что значение процесса в этот момент фиксировано. Это означает, что невзирая на прошлое, будущее процесса будет зависеть только от известного настоящего.

В данный момент растёт востребованность процессов Маркова с доходами. Постановка задачи состоит в том, чтобы выбрать и реализовать алгоритм решения наилучший с точки зрения оптимизации.

Таким образом, актуальность данной темы моей выпускной квалификационной работы объясняется необходимостью разработки программы, реализующей алгоритм управления марковским процессом.

Целью ВКР является реализация алгоритма управления марковской цепью.

Для достижения поставленной цели необходимо решить следующие задачи:

— провести обзор методов решения задач с марковскими процессами;

— выбрать из обозреваемых методов наилучший с точки зрения оптимизации;

— выбрать средства разработки программы;

— разработать программу, реализующую выбранный метод.

Практическая значимость работы заключается в разработке программы, реализующей алгоритм метода управления марковской цепью.

Бакалаврская работа состоит из введения, трёх глав, заключения и списка используемой литературы.

Во введении указывается актуальность темы, а также, описание объекта и предмета исследования, формулируются цели и задачи, необходимые для решения в данной работе.

Первая глава заключается в описании основных понятий и определений цепей Маркова и примеров.

Вторая глава посвящена рассмотрению различных методов решения задач марковских процессов с доходами дальнейшему выбору наилучшего метода.

Третья глава посвящена разработке программы, реализующей алгоритм управления марковской цепью, выбору средств разработки программы, описанию процесса разработки и функциональности программы.

В заключении описываются основные выводы, которые были сделаны в ходе выполнения ВКР.

ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ЦЕПЕЙ МАРКОВА

1.1 Цепь Маркова, матрицы вероятностей перехода

Задаём множество Ω являющееся множеством простых исходов и какое-то множество T, которое будет являться подмножеством для множества, содержащего вещественные числа, проще говоря $T \subset R$.

Случайной функцией будет являться функция, содержащая два аргумента $\omega \in \Omega$ и $t \in T$. Также введём случайную функцию $\xi(w, t)$. При условии, что наш аргумент t обретает смысл во времени $t \geq 0$, то данная случайная функция является случайным процессом. Подытожим, случайный процесс — это случайная функция $\xi(w, t)$, которая определена на множествах $\Omega \times T$, составленная таким образом, что у каждого значения t, которое фиксировано t будет равняться t_0 и принадлежать множеству T (t = t_0 ∈ T) в результате получится случайная величина $\xi(\omega, t_0)$, которая будет определена на множестве Ω. То есть, если значение фиксировано $\omega = \omega_0 \in \Omega$ в результате будет получена функция вещественной переменной $t \in T$: $\xi(\omega_0$,t) $= x \ t$, которая является числовой.[4]

Сечением случайного процесса $\xi(\omega, t)$ является случайная величина $\xi(w)$, которая соответствует устойчивому значению аргумента элемента $t = t_0 \in T$.

Неслучайная функция $x(t)$ называется реализацией случайного процесса, для которой равным может стать процесс, полученный случайно посредством эксперимента.

В ходе работы мы будем использовать обозначение $\xi(t)$ вместо $\xi(\omega, t)$ для сокращение объёма записей.

Итак, возьмём случайный процесс $\xi(t) = \xi_{t-4}$, где ξ — является случайной величиной, которая распределена бернуллиевским законом, имеющая параметр $p = 0{,}7$. Нам нужно изобразить графически и реализовать случайный процесс. Ряд распределения функции ξ будет иметь следующий вид (это обусловлено тем, что она имеет распределение вероятностей Бернулли)[5]:

x_i		
p_i	,2	,7

При условии, что наша случайная величина обнулится, у следующего случайного процесса реализация будет иметь вид $x_1\ \ t\ \ = -4$

В том случае, когда происходит событие $\xi\ =\ 1$, наш случайный процесс будет иметь вид такой функции $x_2\ \ t\ \ = 1 \cdot t - 4 = t - 4$

Наглядно этот процесс можно увидеть на рис. 1.1

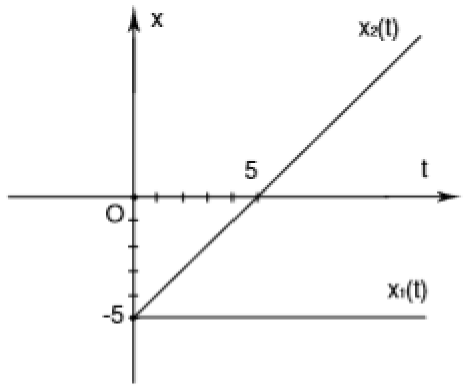

Рис. 1.1 - Асимптоты

Возьмём для рассмотрения некоторую систему, находящуюся в одном из несовместных состояний $\xi(t) = i$, содержащую конечное множество вероятных состояний. $S = \{1,2,\dots,N\}$, т.е. $i \in S$. В ходе работы система изменяется в дискретные моменты времени, которые называют шагами $t = 0,1,2,3,\dots$, из одного состояния $\xi_t = \xi(t)$ в следующее $\xi_{t+1} = \xi(t+1)$.[11]

Простая однородная цепь Маркова с дискретным временем и с фиксированным числом состояний является случайным процессом $\xi(t)$ смены состояний, при условии, что для всех $t \geq 1$ и $i, j,\ i_0,\ i_1, \ldots \in S$ будет выполняться свойство Маркова

$P\{\xi_t = j\ \xi_0 = i_0, \xi_1 =\ i_1, \dots, \xi_{t-1} = i\ \ = P\ \ \xi_t = j\ \xi_{t-1} = i\}$, (1)

когда

$P\{\xi_t = j\ \xi_0 = i_0, \xi_1 = \ i_1, \dots, \xi_{t-1} = i\ \ > 0.$

Это означает, что на следующем этапе система будет переходить из первого состояния во второе, и условная вероятность этого события не будет зависеть ни от настоящего, ни от прошлого состояния системы.

В качестве примера возьмём треугольную пирамиду, которую подбрасывали в течение времени N с интервалом около пяти секунд. Случайное событие $B_i\{\xi\ \ n\ \ = j\}$ является состоянием системы и означает, что на пирамиде выпало j очков.[13]

В этом случае система переходит в один из четырёх вариантов состояний $F\ \ = \{1,2,3,4\}$при этом

$$P\ \ \xi_n = j\ \xi_{n-1} = j\ \ = \frac{1}{4}.$$

Для примера возьмём функцию $\xi(t)$, являющуюся простой цепью Маркова с фиксированным числом состояний и дискретным временем, а множество S — будет обозначать различные её состояния. Количество состояний фиксировано и равняется N. Будем обозначать

$$p_{ij} = P\{\ \xi_t = j\ \xi_{t-1} = i\ \ , \quad (2)$$

где $i, j\ \in\ S, t\ \geq\ 1$, то есть p_{ij} — является условной вероятностью того, что на каком-то шаге t система из состояния i перейдет в состояние j. Такую вероятность p_{ij} называют вероятностью перехода.

Матрица, содержащая все вероятности перехода системы, называется матрицей перехода.

$$P_1 = \begin{matrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{matrix} \quad (3)$$

Отметим $\ {}_{j=1}^{N} p_{ij} = 1$, вследствие того, что состояния системы будут образовывать заполненную группу событий. Матрица переходов из состояния i в состояние j за один шаг обозначим матрицей P_1. Граф состояний широко используется при анализе цепей Маркова, эта геометрическая схема является наиболее наглядной и удобной при решении задач. Внутри круга записывается номер состояния, каждый круг соответствует каждому состоянию марковской цепи. Данные круги являются вершинами графа. Когда из одного состояния в другое возможен одношаговый переход, то между кругами рисуется дуга со стрелкой, направленной в то состояние, в которое переходит система. Рядом с дугой указывается вероятность перехода в состояние.[14]

1.2 Примеры марковских цепей

Когда для каждого момента времени вероятность какого-либо состояния системы в предстоящем является зависимым исключительно от состояния системы в текущий момент и не будет зависеть от способа перехода системы в это состояние, процесс будет называется марковским. К тому же, чтобы считаться марковским, этот процесс должен протекать в физической системе.

Приведём в пример простой вариант случайного процесса Маркова. Точка X случайным образом передвигается по оси абсцисс Ox. В течение двух секунд точка X находится в начале координат $(x = 0)$ в момент времени $t = 0$.

Через две секунды бросается монета, если выпала решка– точка X перемещается на одну единицу длины вправо, если орёл - влево. Ещё через две секунды монета подбрасывается снова и происходит похожее случайное передвижение, и так далее. Случайный процесс с дискретным временем $(t = 0, 1, 2, \dots)$ и счетным множеством состояний $x_0 = 0$; $x_1 = 1$; $x_{-1} = -1$; $x_2 = 2$; … называется процессом «блуждания», то есть процессом изменения точки.[15]

Попробуем показать, что этот процесс является марковским. Действительно, представим себе, что в какой-то момент времени t_0 система находится, например, в состоянии x_1 - на одну единицу правее начала координат. Возможные положения точки через единицу времени будут x_0 и x_2 с вероятностями 1/2 и 1/2; через две единицы - x_{-1}, x_1, x_3 с вероятностями 1/4, ½, 1/4 и так далее. Очевидно, все эти вероятности зависят только от того, где находится точка в данный момент t_0, и совершенно не зависят от того, как она пришла туда.

Представим отличный от предыдущего пример. Имеется техническое устройство X, состоящее из элементов (деталей) типов a и b, обладающих разной долговечностью. Независимо друг от друга эти элементы в различные моменты могут выходить из строя. Правильная работа всех элементов необходима для исправной работы устройства в целом. Для элементов типа a и b параметры по показательному закона различны и равны соответственно λ_a и λ_b, они нужны для определения времени безотказной работы элемента.[17-21]

Неисправный элемент будет немедленно заменён новым, если произошёл отказ устройства, для этого приняты меры для выявления причин отказа.

Время, потребное для восстановления (ремонта) устройства, распределено по показательному закону с параметром μ_a (если вышел из строя элемент типа a) и μ_b(если вышел из строя элемент типа b).

Марковский процесс с непрерывным временем и фиксированным множеством состояний, рассматриваемый в данном примере, является случайным:

x_0 – система функционирует нормально, все элементы работают,

x_1 – система находится в ремонте, неправильно работает элемент a;

x_2 – система находится в ремонте, неправильно работает элемент b;

Что и требовалось доказать, описанный в примере процесс будет обладать марковским свойством. Допустим, система исправна и находится в состоянии x_0 в момент времени t_0.

Момент отказа всех элементов по отдельности в предстоящем не будет зависеть от того, сколько времени он прослужил, потому что время бесперебойной работы элементов является показательным. Поэтому вероятность того, останется ли в будущем система в состоянии x_0 или покинет его в будущем, не будет зависеть от прошлого рассматриваемого процесса.

Теперь можем предположить, что элемент типа a, то есть в момент времени t_0 система находится в состоянии x_1. Вероятность окончания ремонта в любое время после t_0 не будет зависеть от времени начала ремонта и времени, когда был установлены исправно работающие элементы. Исходя из вышесказанного, можем сделать вывод, что процесс является марковским.

Есть важные условия, без присутствия которых процесс нельзя было бы считать марковским. Такими условиями являются показательное распределение времени ремонта, а также показательное распределение времени работы элемента.

Процесс не будет считаться марковским, если время на исправление неполадок будет зависимо от времени, когда это исправление начало осуществляться.

Если представить, что время, когда элемент работает правильно, допустим, распределяется по закону равномерной плотности в промежутке (t_1, t_2), а не по показательному, то мы увидим, что каждая часть безусловно будет работать какое-то время t_1, и может перестать работать в промежутке (t_1, t_2) с похожей вероятностной плотностью.

В этом случае процесс так же не считается марковским, так как наблюдается зависимость поломки элемента в предстоящем от давности установки элемента. И это при условии правильной работы элемента.

Следует учитывать, что для того времени, которое система находится в состоянии x_k, необходимо использовать показательный закон.

Это необходимо при работе с марковскими процессами с недискретным временем. Если предположить, что система присутствует в какой-то момент времени в состоянии x_k и в прошлом также присутствовала в нём, то исходя из понятия марковского процесса, у системы вероятность исчезновения из состояния x_k за какое-то время t должна быть независима от времени, которое система присутствовала в нём. Так как понятие марковского процесса имеет ввиду независимость вероятности появления какого-либо события в предстоящем от прошлого. Параметр при таком распределении зависит от этого состояния.

Такие выводы мы делаем исходя из того, что время у непрерывного марковского процесса распределяется по показательному закону.

$p_1 \; t \;, p_2 \; t \;, \ldots$ это вероятности состояний, которые нужно найти при решении дифференциальных уравнений, которые, в свою очередь, являются описанием марковского процесса.

Это при условии, что у системы имеется количество состояний, которое можно посчитать, и она является физической, а сам процесс недискретным.

Вывод по первой главе

В ходе работы над первой главой выпускной квалификационной работы

Были рассмотрены основные понятия теории марковских цепей. Были рассмотрены такие понятия как:

- случайный процесс;
- цепь Маркова;
- марковский случайный процесс;
- матрица вероятностей переходов.

Основные понятия являются очень важным аспектом и послужат основой для последующей работы над ВКР.

ГЛАВА 2 АНАЛИЗ АЛГОРИТМА УПРАВЛЕНИЯ ЦЕПЬЮ МАРКОВА

2.1 Марковские процессы с доходами

Возьмём в качестве предположения утверждение о том, что система, описанная марковским процессом с N состояниями, приносит доход v_{ij} рублей, когда происходит переход из состояния i в состояние j. Множество доходов системы образует матрицу доходов $R = | v_{ij} | \quad i, \; j \; = \; N$. Тогда можно сформулировать следующую задачу. Если в данный момент система находится в состоянии i, то каким станет средний доход системы n шагов? Этот выигрыш обозначим через

$$v_i \; n \; = \; {}_{j=1}^{n} P_{ij} \; v_{ij} + v_i \; n-1 \; = \; {}_{j=1}^{N} P_{ij} v_{ij} + \; {}_{j=1}^{N} v_i \; n-1 \; p_{ij}, \; (2.1.1)$$

$$i = 1, \ldots, N; j = 1, 2, \ldots$$

Обозначим ${}_{j=1}^{N} P_{ij} v_{ij}$ через q_i. Средний ожидаемый доход при выходе системы из состояния i. Для состояния i q_i будет являться средним ожидаемым доходом.[1-3]

В векторной форме уравнение (2.1.1) можно записать:

$$V \; n \; = q + PV \; n-1 \; , (2.1.2)$$

где $V \; n$ — вектор полных доходов.

Так как матрица P эргодична, то существует

$$\lim_{n\to\infty} P^n = P = \begin{matrix} \lambda_1 \ldots \lambda_m \\ \ldots\ldots\ldots \\ \lambda_1 \ldots \lambda_m \end{matrix}$$

В этом случае матрицу P^kможно представить в виде суммы двух матриц $P^k = (P_k + P)$,

где:

$$\lim_{k\to\infty} P_k = \begin{matrix} 0 \cdots 0 \\ \ldots\ldots \\ 0 \ldots 0 \end{matrix}$$

С учётом этого, уравнение (2.1.2) можно переписать в виде:

$$V\ n\ = V\ 1\ + PV\ 1\ + \cdots + P^{n-1}V\ 1\ =\ P_0 + P\ V\ 1\ +\ P_1 + P\ V\ 1\ + \cdots +\ P_{n-1} + P\ V\ 1 \quad (2.1.3)$$

Или

$$V\ n\ = nPV\ 1\ +\ {}_{k=0}^{n-1} P_k V(1) \quad (2.1.4)$$

Обозначим

$PV\ 1\ =\ \begin{matrix} g \\ \cdots \\ g \end{matrix}$,

где $g =\ {}_{i=1}^{m} \lambda_i V_i(1)$, а $\ {}_{i=1}^{n-1} P_k V(1) = W(n)$

Подставляя W и g в (2.1.4), получаем $V\ n\ = ng + W(n)$ или $V\ n\ = ng + W_i(n)$

Величина W_i называется весом для i-го состояния.

В качестве примера возьмём данные у фабрики столовых приборов.

Специалисты фабрики постоянно работают над разработкой различных дизайнов столовых приборов, чтобы угодить разным требованиям покупателей. Недавно фабрика выпустила новинки. Вследствие этого, она может находится в двух состояниях: S_1- если фабрика выпустила товар, который возымел спрос и S_2 – если фабрика выпустила приборы, которые не пользуются спросом.

Предположим, что если начальное состояние фабрики S_1 и на следующей неделе фирма будет оставаться в этом же состоянии с вероятностью равной 0,5($p_{11} = 0{,}5$), то недельный доход составит 15 условных единиц стоимости (у.е.с.), т.е.$V_{11} = 12$. Если же очередная модель окажется неудачной с вероятностью 0,5 (фабрика перейдёт в состояние S_2), то доход фабрики составит лишь 4 у.е.с. Т.е. $p_{12} = 0{,}5$ и $V_{12} = 7$. Если после модели, не пользующейся спросом, фабрика выпустит удачную модель с вероятностью $p_{21} = 0{,}4$, то доход фирмы составит 4 у.е.с., а если после неудачной модели фирма снова выпустит неудачную с $p_{22} = 0{,}6$, то фирма потерпит убыток в размере 9 у.е.с. Итак, матрица переходных вероятностей имеет вид:

$$P = \begin{matrix} 0{,}5 & 0{,}5 \\ 0{,}4 & 0{,}6 \end{matrix} ,$$

а матрица доходов:

$$V = \begin{matrix} 12 & 4 \\ 4 & -9 \end{matrix}$$

Тогда

$V\ 1\ = \begin{matrix} 0{,}5\cdot 12+0{,}5\cdot 4 \\ 0{,}4\cdot 4-0.6\cdot 9 \end{matrix};V\ 1\ = \begin{matrix} 8 \\ -3{,}8 \end{matrix}$.

Это означает, что если фабрика выпустила удачную модель, то ее средний доход на следующей неделе составит 8 у.е.с., если же она начинает с неудачной модели, то ее ожидаемые потери -3,8 у.е.с. Допустим, что фабрика планирует свою работу на n недель и руководство желает определить ожидаемый доход за это время. Предположим, что $V\ 0\ =0$ Последовательно применяя формулу (2.1.2), найдем $V\ 1\ ,V\ 2\ ,\dots,V(5)$ и эти результаты запишем в таблицу 1.

Таблица 1 Результаты ожидаемого дохода

n	0	1	2	3	4	5
$V_1(n)$	0	8	10,1	11,61	13,061	14,5061
$V_2(n)$	0	-3,8	-2,88	-1,488	-0,0488	1,39512

Тогда уравнения асимптот, будут следующими:

$V_1\ n\ =ng_1+w_1=n+\frac{50}{9};V_2\ n\ =ng_2+w_2=n\cdot 1-\frac{40}{9}.$

В заключении хотелось бы отметить, что асимптоты представляют собой параллельные прямые, причем тангенс угла наклона их равен единице. Такая величина равна среднему доходу за неделю, если бы система функционировала бесконечно долго. Наконец разность $V_1\ n\ -V_2\ n\ =W_1\ n\ -W_2\ n\ =10$ у. е. с. представляет собой дополнительный доход, получаемый фабрикой, если она начинает функционировать в состоянии 1 (т. е. при модели, пользующейся спросом).

2.2 Управляемые марковские процессы с доходами

Если фабрика по производству столовых приборов хочет увеличить спрос на продукцию, ей необходимо использовать рекламу, при нахождении в первом состоянии. Такой способ увеличивает затраты умаляет доходы. Во втором состоянии фабрика пробует повысить затраты на исследования путём повышения вероятности перехода в первое состояние.

Основными будут являться 2 стратегии: в случае с первой, фабрика не тратится на маркетинг и исследования; во второй стратегии фабрика будет использовать оба метода.[7]

У рассматриваемых стратегий матрицы доходов и матрицы вероятностей переходов выглядят так:

$$P_1 = \begin{matrix} p_{11}^1 & p_{12}^1 \\ p_{21}^1 & p_{22}^1 \end{matrix}, R_1 = \begin{matrix} r_{11}^1 & r_{12}^1 \\ r_{21}^1 & r_{22}^1 \end{matrix}, P_2 = \begin{matrix} p_{11}^2 & p_{12}^2 \\ p_{21}^2 & p_{22}^2 \end{matrix}, R_2 = \begin{matrix} r_{11}^2 & r_{12}^2 \\ r_{21}^2 & r_{22}^2 \end{matrix}$$

Чтобы управлять рассматриваемой цепью Маркова, нужно управлять стратегией.

Прямым произведением различных решений $K = K_1 \times K_2 \times \ldots \times K_N$ является пространство различных стратегий K. Здесь для всех состояний от единицы до N поставлено в соответствие какое-то множество K_i альтернативных решений.

$k \in K_i$ – решение, принимаемое, когда в состоянии $i \in S$ система присутствует при нахождении в каком-то состоянии i нескольких множеств вероятностей перехода p_{ij}^l , $l = 1, k_l$. В случае K_i $= 1$ получается марковская цепь, которая не может управляться. Когда эти условия соблюдаются система ведёт себя следующим образом:

— при помощи вероятности будет определяться состояние системы в будущем;

— r_i^k будет называться доходом, который к ней поступает.

Для каждого $i \in S$ и $k \in K_i$ доход будет ограничиваться исходя из того, что выбирается определённое решение в каком-то состоянии из определённого перечня вероятностей перехода.

Так же следует заметить, что:

$\sum_{j=1}^{N} p_{ij}^{k} = 1, p_{ij}^{k} \geq 0$ для каждого $i, j \in S$ и $k \in K_i$.

Если случайная величина имеет зависимость при выборе решения от её местонахождения в начале, то доход, который система имеет в результате итераций, считается величиной случайной. Принято считать, что буквы (K, P, r) могут описать марковскую управляемую цепь. Здесь последняя буква определяет доходы, вторая — переходные вероятности, а первая означает принимаемые действия относительно этой системы.

Если задать стратегию, используя некоторые решения для каждого временного момента $t = 1, 2, \ldots, \pi, \ldots$, мы можем взять решения для какого-то состояния i , при условии присутствия системы в текущем моменте.

При оценке эффективности управления мы будем рассматривать какое-то множество решений и оценивать весь доход, а также доход усреднённый в определённое время. В первом случае мы будем работать с дискретным временем, а во втором — с непрерывным. То решение, которое мы выберем в определённое время, является управлением цепью.

Для момента времени π мы принимаем решение для какого-либо состояния, обозначаемого $i \in S$. Ведём понятие стратегии — это такой порядок решений $\pi = f_1, \ldots, f_n, \ldots$, где n-ая функция является вектором вида, а аргумент этого вектора будет значиться решением, которое принимается при управлении.

Если стратегию записывают как f^{∞}, она будет являться стационарной. При рассмотрении стратегии Маркова f_n, называемое решением, которым управляют во всех состояниях, данное решение не должно отвлекаться на прошлые состояния, а также на решения, которые были в них взяты. Когда решение находится в зависимости от π, то есть от текущего момента, мы можем называть стратегию марковской.

$\pi_n = (f_1, \ldots, f_n)$ в примере будет обозначать нефиксированную стратегию, а в какой-то момент времени мы возьмём непроизвольную стратегию, которую обозначим $\pi = (f_1, \ldots, f_n, \ldots)$. Предположим, что присутствие системы в состоянии $j \in S$ определяется некоторой вероятностью p_{ij}^{k}, в этом обозначении индекс k означает $f_n(i)$. Это соблюдается при условии, что система присутствует в $(\pi + 1)$-ом моменте.

Исходя из этого наша матрица вероятностей перехода будет выглядеть следующим образом:

$$P = P(f_n) = \begin{matrix} p_{11}^{f_n(1)} & \cdots & p_{1N}^{f_n(1)} \\ \cdots & \cdots & \cdots \\ p_{N1}^{f_n(N)} & \cdots & p_{NN}^{f_n(N)} \end{matrix}$$

Марковская цепь станет доступна в результате выбора стабильной стратегии, у которой $P = P\ \ f_n\ \ , n = 1,2,3, \ldots$ является переходными матрицами.

Из вышесказанного можно сделать выводы:

Первый вывод. У каждого дискретного времени имеется наилучшая стратегия Маркова.

Второй вывод. У каждого непрерывного времени имеется наилучшая стратегия стационарная.

У фиксированного времени наилучшей стратегией будет та, которая дискретна, а также будет являться марковской, и при выборе решения она может быть зависима от текущего момента выбора решения. А при непрерывном времени наилучшая стратегия будет зависеть исключительно от того, где находится система, но не будет зависеть от какого-либо временного показателя и от предшествующих выбранных порядков состояний и избранных решений.

Далее, я рассмотрю модель с помощью рекуррентного метода. Выше было изучено асимптотическое поведение полного ожидаемого дохода $V(n)$ для процессов бесконечной длительности.

2.3 Рекуррентный метод

Применю метод отыскания оптимальной стратегии на каждом шаге такого процесса. Для этого несколько видоизменю задачу о фабрике по созданию столовых приборов.

Если фабрика хочет увеличить свой доход, это действие может поменять вероятности. В случае, когда фабрика выпускает модель, пользующуюся спросом, руководство фабрики может заняться дополнительным маркетингом, дабы спрос повысился ещё сильнее. Так как маркетинг требует денег, траты на него снизят ожидаемый доход.

Итак, для примера возьмём переходные вероятности из первого состояния $P_{1j} = [0{,}6\ 0{,}4]$, они будут таковыми, если фабрика решит воспользоваться маркетингом. $r_{1j} = [5\ 5]$ в этом примере будет являться расходным распределением.

Фабрика может использовать маркетинг или не делать этого, когда находится в первом состоянии. Два этих варианта развития событий мы назовём первой и второй стратегией соответственно. Отметим стратегии в каждом состоянии индексом $k(k = 1{,}2)$. Таким образом, для состояния 1 произвелись следующие вычисления:

$$P_{1j}^{(1)} = \ 0{,}5\ 0{,}5\ , r_{1j}^{(1)} = \ 12\ 4\ , P_{1j}^{(2)} = \ 0{,}6\ 0{,}4\ , r_{1j}^{(2)} = \ 5\ 5\ .$$

В состоянии 2 также возможны несколько вариантов стратегий. Например, можно увеличить затраты на исследования, что повышает вероятность получения удачной модели, но при этом возрастает и стоимость пребывания системы в данном состоянии (уменьшаются доходы). Эта стратегия (обозначим ее через 2) дает следующее распределение вероятностей состояний и доходов:

$P_{2j}^{2} = \ 0{,}7\ 0{,}3$ и $r_{2j}^{2} = \ 2\ -22\ .$

Число стратегий в каждом состоянии может быть различным, но должно быть конечным. Стратегии руководства фирмы приведены в таблице 2.

Таблица 2 Стратегии руководства фирмы

Состояние i	Стратегия k	Доходы				q_i^k
		p_{i1}^k	p_{i2}^k	r_{i1}^k	r_{i2}^k	
Успешная модель	Без рекламы	0,5	0,5	12	4	8
	С рекламой	0,6	0,6	5	5	6
Неуспешная модель	Без проведения исследований	0,4	0,6	4	-9	-3,8
	С проведением исследований	0,7	0,3	2	-22	-5,2

Величина $q_i^k = \sum_{i=1}^{N} P_{ij}^k r_{ij}^k$ является ожидаемым доходом за один переход из состояния i при выборе стратегии k.

Предположим, что в распоряжении руководства осталось n недель до закрытия производства. Определим $d_i(n)$ как номер стратегии, которая будет использоваться на n шаге, если система находится в состоянии i. Назовем $d_i(n)$ решением в состоянии $i \cdot n$-го шага. Очевидно, стратегии руководства будут определены, если для всех i, n заданы $d_i(n)$. Оптимальным будет такое поведение, которое максимизирует полный ожидаемый доход для всех i и n

Пусть $V_i(n)$ есть полный ожидаемый доход за n шагов при оптимальном поведении, если система начинает функционировать из состояния i. Тогда для любого n справедливо следующее основное рекуррентное соотношение динамического программирования:

$$V_i(n+1) = \max_k \left[\sum_{j=1}^{N} P_{ij}^k \left(r_{ij}^k + V_j(n)\right)\right], n = 1, \dots, N \qquad (2.3.1)$$

В соответствии с (2.3.1) на каждом шаге n выбирают такую стратегию k, которая обеспечивает максимальный доход за оставшееся до окончания процесса число шагов.

Для того чтобы применить эти соотношения, нужно задаться начальными значениями доходов $V(0)$. Примем $V_1\ 0\ = V_2\ 0\ = 0$. Проиллюстрируем ход вычислений на нашем примере.

Учитывая, что $V_1\ 0\ = 0$, находим $V_1\ 1$: $V_1\ 1\ = \max_k q_1^k$

Так как $q_1^1 = 8, q_1^2 = 6$, то в состоянии 1 лучше использовать первую стратегию. При этом $V_1\ 1\ = 8$.

Аналогично, $V_2\ 1\ = \max_k q_2^k$.

Так как $q_2^1 = -3{,}8$, а $q_2^2 = -5{,}2$, то и в состоянии 2 лучшей стратегией оказывается первая; $V_2\ 1\ = -3{,}2$. Вычислив теперь $V_1\ 1$ и $V_2\ 1$, можно аналогично определить $V_i\ 2\ \ (i = 1{,}2)$ и стратегии второго шага $d_1\ 2\ , d_2(2)$. Результаты вычислений представлены в таблице 3.

Таблица 3 Решение задачи фабрики рекуррентным методом

n	0	1	2	3	4
$V_1\ n$	0	8	11,64	15,164	18,6764
$V_2\ n$	0	-3,8	-0,74	2,276	6,2326
$d_1\ n$	-	1	2	2	2
$d_2\ n$	-	1	2	2	2

Расчёты:

	P		R		V(1)	V(2)	V(3)	V(4)
Стр.1	0,5	0,5	12	4	8	10,1	13,45	16,945
Стр.2	0,8	0,2	6	6	6	11,64	15,164	18,6764
Стр.1	0,4	0,6	4	-9	-3,8	-2,88	0,412	3,9012
Стр.2	0,7	0,3	2	-22	-5,2	-0,74	2,726	6,2326

Заметим, что для $n = 2{,}3, \ldots$ в каждом состоянии следует предпочесть вторую стратегию. Это значит, что для корпорации выгоднее пользоваться рекламой и проводить исследования, несмотря на увеличение расходов.

Рассмотренный метод исследования процессов последовательного принятия решений называется рекуррентным. Он пригоден для анализа процессов ограниченной длительности. Если же число шагов до окончания процесса велико (или бесконечно), то применять этот метод не эффективно, так как требуется последовательно вычислять функции $V_i(n)$ для всех $n = 1,2,\ldots$

Далее рассмотрим метод, предназначенный специально для анализа процессов бесконечной длительности и позволяющий установить их асимптотические свойства.

2.4 Метод итераций

Рассмотрим эргодический марковский процесс с доходами и N состояниями бесконечной длительности.

Этот процесс характеризуется величиной среднего дохода за единицу времени, или прибылью g. Так как процесс эргодичен, то предельные вероятности $\pi_i (i = 1, \ldots, N)$ не зависят от начального состояния, а прибыль определяется соотношением:

$$g = \sum_{i=1}^{N} \pi_i q_i$$

где q_i — непосредственно ожидаемый доход в состоянии i.

Оптимальным станет решение, которое будет максимизировать прибыль (средний доход за 1 переход). Всякое решение будем определять вектором d, компоненты которого представляют собой номера стратегий на соответствующих шагах процесса.

Рассмотрим предложенный Р. Ховардом способ нахождения оптимального решения за небольшое число итераций. Согласно величины $V_i(n)$ удовлетворяют рекуррентному соотношению:

$$V_i(n) = q_i + \sum_{j=1}^{N} P_{ij} V_j(n-1), i = 1, N, n = 1,2, \ldots$$

С другой стороны, для эргодических процессов $V_i(n)$ имеет следующий асимптотический вид:

$$V_i(n) = ng_i + W_i, i = 1, N$$

Рассмотрим асимптотическое поведение системы. Для этого переносим $V_i(n)$ в уравнение

$$ng_i + W_i = q_i + \sum_{j=1}^{N} P_{ij}\left[(n-1)g_j + W_j\right], i = 1, N,$$

$$ng_i + W_i = q_i + (n-1)g_i \sum_{j=1}^{N} P_{ij} + \sum_{j=1}^{N} w_j$$

Так как $\sum_{j=1}^{N} P_{ij} = 1$, эти уравнения преобразуются к виду:

$$g_i + W_i = q_i + \sum_{j=1}^{N} P_{ij} W_j, i = 1, N$$

Итак, мы получили систему из N линейных уравнений, связывающих величины W_i и искомую прибыль g с матрицами вероятностей $P = \| P_{ij} \|, i, j = N$ и доходов $R = \| r_{ij} \|, i, j = N$

Для определения N неизвестных V_i и g необходимо $(N + 1)$ уравнение, а есть только N уравнений.

Для того, чтобы выйти из этого затруднения, необходимо любой из весов W_i (например $W_n = 0$) положить равным нулю, а остальные веса определить относительно данного.

Правомочность такого способа определения базируется на следующем свойстве: если прибавить какую-то константу ко всем W_i, то вид уравнения не изменится. Действительно,

$g + (W_i + a) = q_i + \sum_{j=1}^{N} P_{ij}(W_i + a)$ и

$g + W_i = q_i + \sum_{j=1}^{N} P_{ij} W_j$,

а потому неизвестные $W_i (i = 1, \dots, N)$ могут быть определены с точностью до некоторой константы. Эти веса называются относительными.

Если i-е уравнение умножить на π_i — предельную вероятность i-го состояния, а затем просуммировать по всем i, то

$$g\sum_{i=1}^{N}\pi_i + \sum_{i=1}^{N}\pi_i W_i = \sum_{i=1}^{N}\pi_i q_i + \sum_{i=1}^{N}\sum_{i=1}^{N}\pi_i P_{ij} W_i$$

Сравнивая, видим, что эти выражения эквивалентны.

Выше показано, что при использовании оптимальных решений вплоть до π-го шага, оптимальная стратегия на следующем $(\pi+1)$-м шаге может быть найдена из условий максимума по переменной k выражения

$$q_i^k + \sum_{i=1}^{N} P_{ij} V_i(n)$$

При больших n, заменив $V_j(n)$, на $ng_j + w_j$, получим критерий максимизации в виде

$q_i^k + \sum_{j=1}^{N} P_{ij}^k (ng + W_j),\ i = 1, N$ (2.4.1)

Так как:

$\sum_{j=1}^{N} P_{ij}^k = 1,$

то слагаемое $\sum_{j=1}^{N} ngP_{ij}^k$ от k не зависит и может быть удалено из критерия.

Таким образом, если система находится в состоянии i, для определения оптимальной стратегии k необходимо найти:

$\max_k \{q_i^k + \sum_{j=1}^{N} P_{ij}^k W_j\}$ (2.4.2)

Причем в этом выражении можно использовать относительные веса, полученные из решения уравнений на предыдущем шаге.

Итак, одна итерация (цикл) метода Ховарда состоит из двух этапов.

1) Определение весов. Используя P_{ij} и q_i решения данного шага, находим прибыль g и относительные веса W_i из системы уравнений

$g + W_i = q_i + \sum_{j=1}^{N} P_{ij} W_j,\ i = 1, N,$

положив, например, $W_N = 0$

2) Улучшение решения. Для каждого состояния i,используя относительные веса предыдущего решения, находим стратегию k',которая максимизирует критерий $(q_i^k + \sum_{i=1}^{N} P_{ij}^k W_i)$

Затем приняв эту стратегию за новое решение в i-состоянии, заменим q_i^k на $q_i^{k\prime}$, а P_{ij}^k на $P_{ij}^{k\prime}$ и возвратимся к этапу «определение весов».

Итерационный процесс можно начинать с любого этапа. Если исходным этапом избрать первый, то нужно подобрать начальное решение, если же начинать со второго этапа, то необходимо задать набор начальных весов. Если предварительной информации относительно выбора специального начального решения нет, то можно начать процесс, положив все до $W_i = 0$. Тогда для каждого состояния i будет найдена стратегия k', которая максимизирует величину q_i^k — непосредственный доход. Затем в работу вступает этап определения весов (с учетом решения $d_i = k$) и начинается итерационный цикл.

Сформулируем правило прекращения итераций: оптимальное решение будет найдено (а прибыль g максимизирована), когда совпадут решения двух последовательных итераций.

Чтобы избежать возможности зацикливания при наличии нескольких одинаково хороших стратегий в некотором состоянии потребуем, чтобы решение предыдущего шага d_i оставалось неизменным, если величина критерия для него такая же, как и для некоторых других стратегий вновь определяемого решения.

Отметим основные свойства итерационного метода:

1. Определение оптимального решения процесса марковского типа сводится к решению системы линейных уравнений с последующим сравнением.

2. Каждое следующее решение, которое находят с помощью итерационного цикла, имеет большую прибыль, чем предыдущее.

3. Итерационный процесс будет закончен при получении решения, обеспечивающего наибольшую допустимую прибыль.

Рассмотрим задачу о фабрике по производству столовых приборов. Здесь существует два состояния и две стратегии в каждом из них. Руководству необходимо определить, какого из решений нужно придерживаться задолго до остановки производства, чтобы получить максимальную прибыль.

Так как мы вначале не знаем, какое решение лучше, положим

$$W_1 = W_2 = 0$$

Для системы в состоянии 1 ($i = 1$), оптимальная стратегия определится из условия $\max_{k=1,2} q_1^k = \max \{8\ 6\}$

Аналогично, если система была в состоянии 2, то номер оптимальной стратегии k определится соотношением:

$$\max_{k=1,2} q_2^k = \max\{-3{,}8\ -5{,}2\}$$

Итак, первое решение руководства фирмы состоит в выборе первой стратегии в обоих состояниях 1 и 2. При этом:

$$d = \begin{matrix} 1 \\ 1 \end{matrix};\ P = \begin{matrix} 0{,}5 & 0{,}5 \\ 0{,}4 & 0{,}6 \end{matrix};\ q = \begin{matrix} 8 \\ -3{,}8 \end{matrix}.$$

Далее переходим к этапу определения весов. Получим:

$$g + W_1 = 8 + 0{,}5W_1 + 0{,}5W_2;\ g + W_2 = -3{,}8 + 0{,}4W_1 + 0{,}6W_2$$

Полагая $W_2 = 0$ и решая эти уравнения, получаем $g = 1{,}45$; $W_1 = 13{,}11$. Выполним далее этап улучшения решения и результаты занесем в таблицу 4.

Таблица 4 Результаты решения

Состояние i	Стратегия k	Критерий ($q_i^k + \sum_{j=1}^{N} P_{ij}^k W_j$)
1	1	$8 + 0{,}5 \cdot 10 + 0{,}5 \cdot 0 = 13$
	2	$6 + 0{,}8 \cdot 10 + 0{,}2 \cdot 0 = 14 \leftarrow$
2	1	$-3{,}8 + 0{,}4 \cdot 10 + 0{,}6 \cdot 0 = 0{,}2$
	2	$-5{,}2 + 0{,}7 \cdot 10 + 0{,}3 \cdot 0 = 1{,}8 \leftarrow$

Как видим, вторая стратегия в каждом состоянии приводит к большему значению величины критерия $q_i^k + \sum_j^N P_{ij}^{(k)} W_i$), чем первая.

Поскольку нет полной уверенности, что новое решение будет наилучшим, продолжим процесс итераций. Для решения $d = \begin{matrix} 2 \\ 2 \end{matrix}$, имеем:

$$P_2 = \begin{matrix} 0{,}8 & 0{,}2 \\ 0{,}7 & 0{,}3 \end{matrix}\quad q_2 = \begin{matrix} 6 \\ -5{,}2 \end{matrix}$$

Переходя к этапу определения весов, запишем:

$$g + W_1 = q_1^2 + \sum_{j=1}^{2} P_1^{(2)} W_j = 6 + 0{,}8\, W_1 + 0{,}2 W_2;,$$

$$g + W_2 = q_2^2 + \sum_{j=1}^{2} P_2^{(2)} W_j = -5{,}2 + 0{,}7\, W_1 + 0{,}3 W_2;$$

Полагая $W_2 = 0$, находим $g = 3{,}51$, $W_1 = 12{,}44;\ W_2 = 0$

Таким образом, прибыль процесса при решении $d^{-T} = [2\ 2]$ увеличивается более чем в два раза по сравнению с прибылью исходного решения. Обращаемся к процедуре улучшения решения. Однако в силу того, что относительные веса случайно совпали с весами предыдущей итерации, вычисления повторяются. В итоге вновь получим решение $d = [2\ 2]$, а поскольку оно совпадает с предыдущим, то должно быть оптимальным.[10]

Итак, фирме следует придерживаться второй стратегии в любом состоянии, и тогда в среднем ее доход за неделю составит 3,51 у.е.с.

Величина $W_1 - W_2 = 12{,}44$ составляет ту сумму, которую фирма может позволить себе истратить на исследования по изготовлению удачных моделей.

Итерационный метод применялся только к эргодическим процессам бесконечно большой длительности (или с очень большим числом шагов до окончания).[11]

Приведём обоснование итерационного метода.

Предположим, что мы нашли некоторое решение A и в результате его улучшения было получено В, отличное от A. Тогда нужно показать, что $g^B > g^A$, где g^A, g^B — прибыль, полученная при решениях A и В соответственно. Поскольку решение В получено в результате улучшения А, то:

$$q_i^B + \sum_{j=1}^{N} P_{ij}^B W_j^A \geq q_i^A + \sum_{j=1}^{N} P_{ij}^A W_j^A,\ i = 1, N \quad (2.5.1)$$

Обозначим:

$$\gamma_i = q_i^B + \sum_{j=1}^{N} P_{ij}^B W_j^A - q_i^A - \sum_{j=1}^{N} P_{ij}^A W_j^A \quad (2.5.2)$$

Используя уравнение для каждого из решений А и В, получаем:

$$g^B + W_i^B = q_i^B + \sum_{j=1}^{N} P_{ij}^B W_j^B,\ i = 1, N \quad (2.5.3)$$

$$g^A + W_i^A = q_i^A + \sum_{j=1}^{N} P_{ij}^A W_j^A, \quad (2.5.4)$$

При вычитании, получим

$$g^B - g^A + W_i^B - W_i^A = q_i^B - q_i^A + \sum_{j=1}^{N} P_{ij}^B W_j^B - \sum_{j=1}^{N} P_{ij}^A W_j^A \quad (2.5.5)$$

Если определить $q_i^B - q_i^A$ из (2.5.2) и подставить в (2.5.5), то получим:

$$g^B - g^A + W_i^B - W_i^A = \gamma_i - \sum_{j=1}^{N} P_{ij}^B W_j^A + \sum_{j=1}^{N} P_{ij}^A W_j^A + \sum_{j=1}^{N} P_{ij}^B W_j^B - \sum_{j=1}^{N} P_{ij}^A W_j^A$$

Или:

$$g^B - g^A + W_i^B - W_i^A = \gamma_i + \sum_{j=1}^{N} P_{ij}^B (W_i^B - W_i^A) \quad (2.5.6)$$

Пусть:

$$\Delta g = g^B - g^A;\ \Delta W_i^B = W_i^B - W_i^A;\ \Delta W_j = W_i^B - W_i^B,$$

Тогда:

$$\Delta g + \Delta W_i = \gamma_i + \sum_{j=1}^{N} P_{ij}^B \Delta W_j, i - 1, N \quad (2.5.7)$$

Уравнения по форме совпадают с той лишь разницей, что они записаны для приращений прибылей и весов, а не их абсолютных величин. Точно так же, как прибыль q, полученная из уравнений, была равна $g = \sum_{i=1}^{N} q_i \pi_i$ величину Δg можно представить в виде $\Delta g = \sum_{i=1}^{N} \pi_i \Delta q_i = \sum_{i=1}^{N} \pi_i^B \Delta \gamma_i$ (2.5.8)

где : π_i^B— предельные вероятности состояний при решении В.

В ходе исследования итерационного метода можно сделать вывод, что все $\pi_i^B \geq 0$ и все $\gamma_i \geq 0$, то и $\Delta g = g^B - g^A > 0$. В частности g^B будет больше, чем g^A, если хотя бы для одного состояния при решении В может быть получено увеличение критерия (2.5.1).

Вывод по второй главе

Во второй главе выпускной квалификационной работы были рассмотрены марковские процессы с доходами.

Для них были рассмотрены два метода: рекуррентный и итерационный. В ходе исследования данных методов было решено выбрать наиболее оптимальный для работы с неограниченным количеством этапов (шагов) процесса. Данный метод будет использовать свойство эргодичности марковской цепи и будет заключаться в последовательном уточнении решения путем повторных расчетов (итераций). При таких уточнениях мы найдём решение, которое будет обеспечивать в среднем максимум дохода при большом числе шагов. А также, оно не станет зависеть от того, на каком шаге мы производим оценку оптимальной стратегии, то есть будет являться справедливым для всего процесса, в независимости от номера шага. Неотъемлемым достоинством данного метода ещё является, кроме того, и то, что он дает возможность определить момент прекращения дальнейших уточнений.

Главным отличием итерационного метода от рекуррентного, который был рассмотрен в главе, заключается в том, что в этом случае мы используем матрицу предельных (финальных) вероятностей, где в силу свойства эргодичности на всех шагах процесса переходные вероятности постоянны. Так как матрица доходов состоит также из постоянных величин, которые не зависят от n, мы можем предположить, что с ростом n общее количество доходов будет возрастать линейно.

Также, в главе было дано обоснование итерационного метода, которое пригодится при отладке реализованной программы.

ГЛАВА 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА УПРАВЛЕНИЯ МАРКОВСКОЙ ЦЕПЬЮ

3.1 О начальных данных и выборе средств программной реализации

Как уже было сказано в главе 2, оптимальным мы будем называть то решение, которое максимизирует прибыль.

Сложность реализации данного алгоритма состоит в том, что мы имеем дело с эргодическим марковским процессом с доходами и N состояниями бесконечной длительности. Это значит, что предельные вероятности не будут зависеть от начального состояния, а прибыль будет определяться соотношением $g = \sum_{i=1}^{N} \pi_i \, q_i$, где q_i ожидаемый доход в состоянии i.[16]

За основу программы я возьму уже рассмотренную во второй главе формулу (2.4.2). Чтобы её использовать, мне нужны следующие данные:

1) вектор d, который включает в себя номера стратегий на шагах процесса, которые им соответствуют;

2) матрица вероятностей P;

3) q_i непосредственно ожидаемый доход в состоянии i.

Также, необходимо определить веса, для этого один из W_i мы будем вычислять, положив остальные нулями. Это мы будем делать до тех пор, пока не определим все веса.

С помощью этих исходных данных в дальнейшем была разработана программа.

Для программной реализации необходимо выбрать среду разработки. Было решено писать программу на языке программирования С++.[8]

Для данного языка являются популярными следующие среды разработки:

- Microsoft Visual Studio;
- NetBeans;
- Eclipse;
- Qt Creator;
- Code::Blocks;

- Geany;
- Dev-C++.

Исходя из личных предпочтений и из соображений удобства я выбрала среду разработки Code::Blocks.

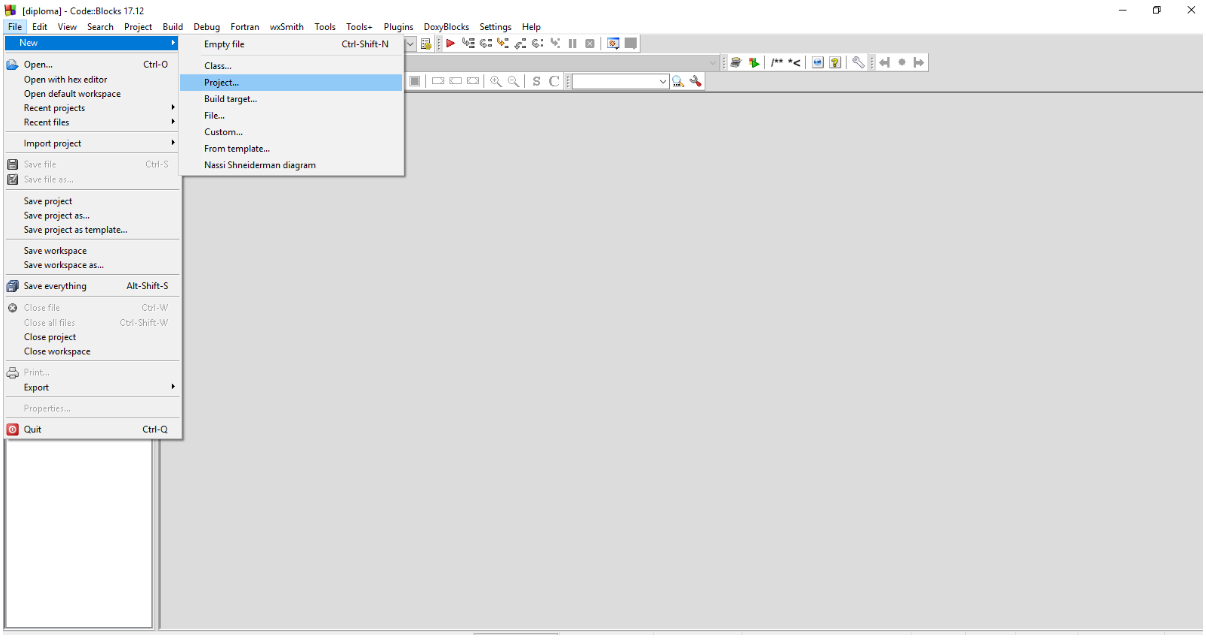

Рис. 1.2 – Начало работы в Code::Blocks

Прочие среды разработки были испробованы мной в ходе обучения в университете, поэтому я пришла к выводу, что для данной работы, выбранная мной среда является наиболее оптимальной.[9]

Также, необходимо описать системные ресурсы, используемые при разработке программы: 200 Мб RAM, Windows 64-bit: Windows 10,

3.2 Описание алгоритма работы программы

Алгоритмы были реализованы в Code::Blocks – это интегрированная среда разработки для языка программирования C++. Code::Blocks была разработана компанией The Code::Blocks team.

Для работы программы нам необходимы следующие встроенные библиотеки:

- math.h;
- vector;

- cstdlib;
- array;
- exception;
- fstream.

Встроенные библиотеки в C++ являются коллекциями функций и классов, которые были написаны на базовом языке. Они поддерживают ряд основных контейнеров, а также функций для того, чтобы работать с этими контейнерами.

Далее нам нужно создать массивы со значениями матриц доходов и расходов, а также создать переменные, в которых будут храниться веса. И переменные, хранящие временные значения функций максимизации доходов и минимизации расходов.

Наш итерационный алгоритм состоит в следующих действиях:

1. Задать $\varepsilon > 0$;
2. Создание класса для решения методом итераций;
3. Функция для поиска весов;
4. Функция проверки на ошибки, то есть не корректный ввод входных данных;
5. Функция для вывода необходимых выходных данных, таких как номер наилучшей стратегии;
6. Реализация метода итераций;
7. Класс для хранения промежуточных значений функции максимизации.

Создание класса для решения методом итераций.

В этом классе мы будем использовать свойства марковских процессов, описанных в главе 2.

В следующем шаге мы будем искать необходимые веса для последующих вычислений.

Класс, описывающий метод итераций будет включать в себя два спецификатора доступа:

— private – в нём я указываю все методы и свойства класса, которые не будут меняться за пределами самого класса;

— public – в нём я указываю все методы и свойства класса, которые могут и будут использоваться и изменяться в других блоках программы в ходе её работы.

У первого спецификатора будут находиться начальные данные, которые не должны измениться после работы программы, например, такие как матрицы доходов и расходов, а также матрица вероятностей переходов.

Второй спецификатор будет включать в себя данные, которые программа при работе будет менять. Например, веса, которые будут заново рассчитываться с каждой новой итерацией, значения критериев, рассчитываемых на каждой итерации до последней.

Функция для поиска весов будет считать их, решая системы уравнений, где все веса критериев будут приниматься за 0, кроме одного, он и будет вычислен. Таким образом будут рассчитаны все веса для критериев на одной итерации.

Задать $\varepsilon > 0$ мне поможет функция rand(), использующая стандартную библиотеку именуемую cstdlib.

Также необходимо проверить входные данные на правильность ввода, если данные были введены правильно, программа продолжит свою работу с ними. Если же пользователь ввёл исходные данные неверно, например, вместо цифры случайно написал букву или поставил лишний пробел, тогда программа остановит свою работу и сообщит пользователю об ошибке. Будет учитываться ещё и тип ошибки, программа выведет сообщение, в котором будет указано, какую ошибку он допустил.

Класс для хранения промежуточных значений функции максимизации будет использовать стандартную библиотеку fstream. Она предназначена для манипуляций с данными из файла. С помощью данной библиотеки мы создадим файл, в который будут записываться значения функции максимизации для каждой итерации, чтобы потом использовать эти данные для определения наилучшей стратегии. Значения функции максимизации будут считываться из файла поэтапно для их последующего сравнения между собой.

Стандартные библиотеки vector и array будут использоваться при работе с матрицами доходов, расходов и переходных вероятностей.

Удобство стандартной библиотеки vector состоит в том, что у её хранилища осуществляется автоматическая обработка данных, а так же её данные могут изменяться в размере.

Цикл, записывающий матрицы, вводимые пользователем, будет иметь две составляющих – значение столбца, обозначаемое буквой n и значение строки, обозначаемый буквой m, то есть индексов элемента. Ввод будет осуществляться поэлементно, ровно так же, как и вывод матриц на экран.

Блок-схема цикла, осуществляющего ввод, представлена на рис. 1.3

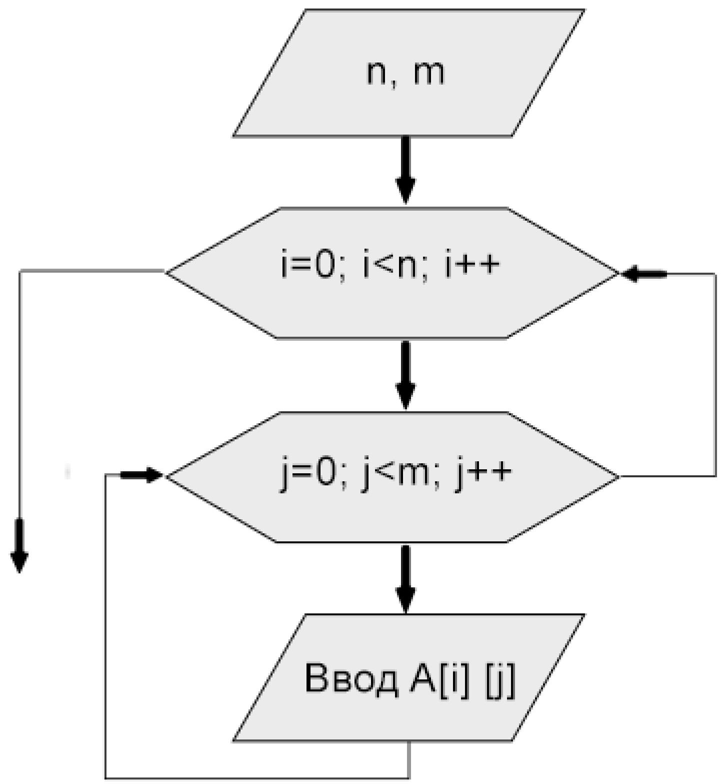

Рис. 1.3 – Ввод элементов матриц

Примерная блок-схема алгоритма программы представлена на рисунке 1.4. На ней видно итерационный цикл.

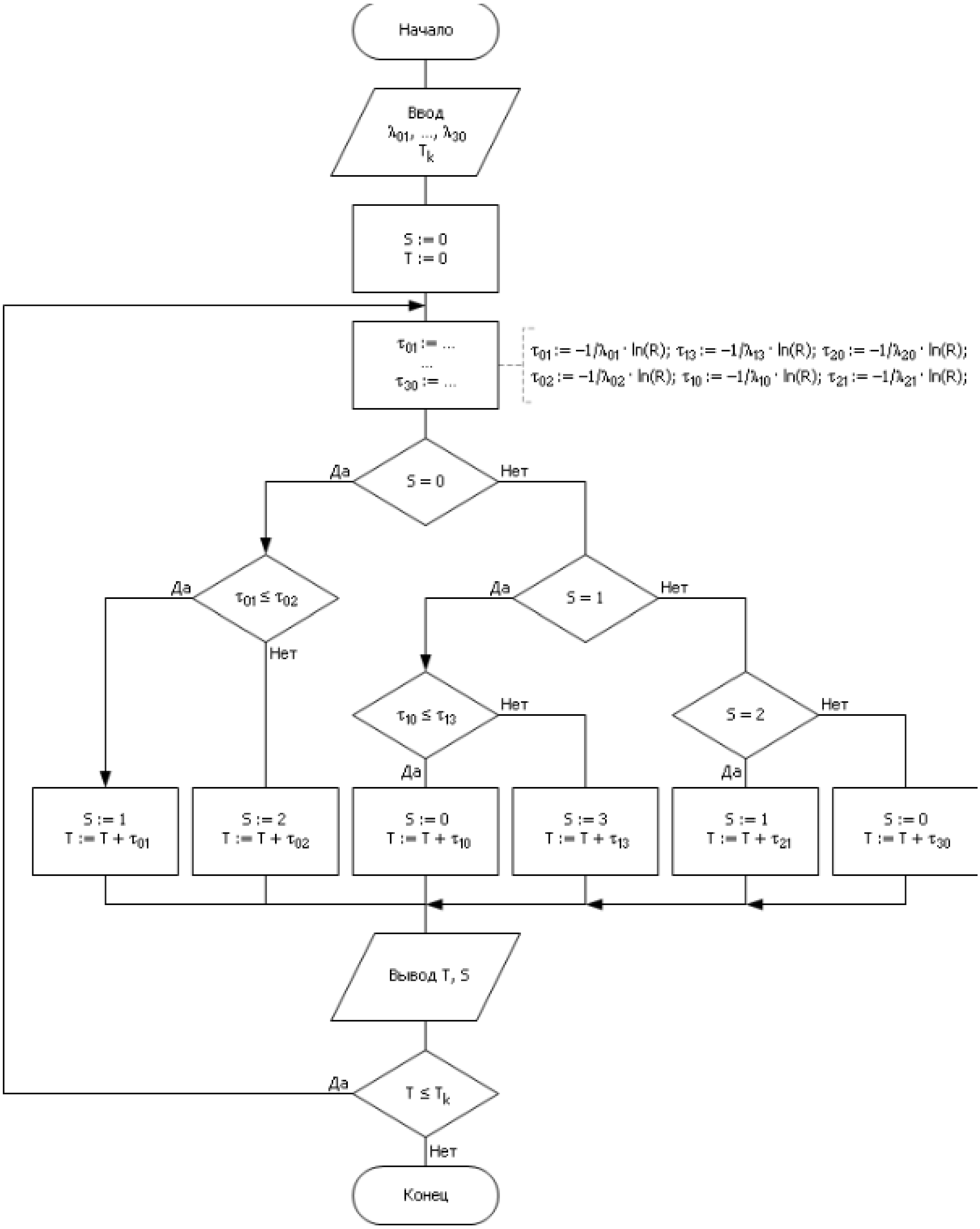

Рис. 1.4 – Пример блок-схемы

Пример результата работы моей программы приведён на рисунке 1.3

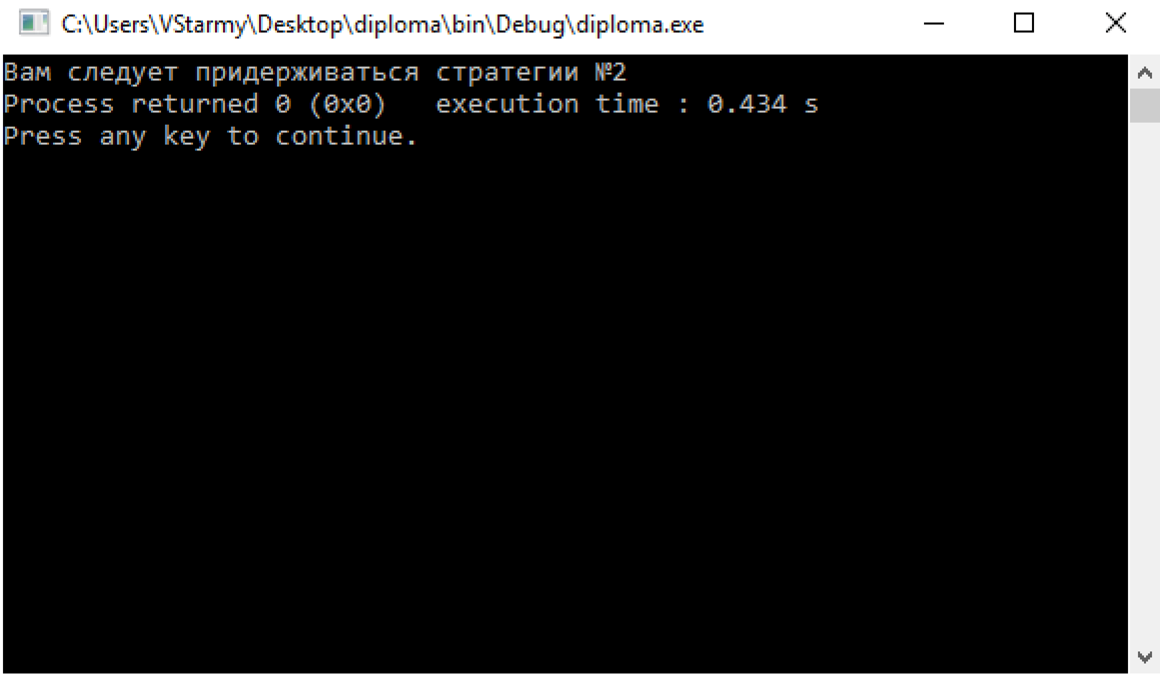

Рис. 1.5 – Результат работы программы

На рисунке 1.3 слова «Вам следует придерживаться стратегии №2» означают, что при имеющихся входных данных, организации следует использовать вторую стратегию в любом состоянии системы для увеличения доходов и уменьшения расходов.

Так была найдена оптимальная стратегия для распределения финансов организации.

ЗАКЛЮЧЕНИЕ

Цель данной выпускной квалификационной работы состояла в реализации алгоритма управления марковской цепью. Для её реализации были выполнены задачи.

Повышение доходов и уменьшение расходов – важная задача для любого бизнеса. И одним из способов выполнения этой задачи является создание программы, реализующей алгоритм управления марковским процессом.

В ходе работы над решением поставленной задачи и проведения исследования мной были получены следующие результаты:

— проведён обзор методов решения задач с марковскими процессами;

— выбран из обозреваемых методов наилучший с точки зрения оптимизации;

— выбраны средства разработки программы;

— разработана программа, реализующая выбранный метод.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Учебники и учебные пособия

1. Ванько В.И., Ермошина О.В., Кувыркин Г.Н. Вариационное исчисление и оптимальное управление. – М.: МГТУ им. Баумана, 1999
2. Городецкий С.Ю., Гришагин В.А. Нелинейное программирование и многоэкстремальная оптимизация. – Н.Новгород: ННГУ, 2007
3. Галлеев Э.М. Оптимизация: теория, примеры, задачи. Учебное пособие. – М.: Элиториал УРСС, 2000
4. Дынкин Е. Б. Основания теории марковских процессов. — М.: Физматгиз, 1959
5. Дынкин Е. Б. Управляемые случайные последовательности // Теория вероятн. И её примен. — 1965. — Т. X, в. 1. — С. 3-18
6. Дынкин Е. Б. Марковские процессы. — М.: Физматгиз, 1963
7. Измайлов А.Ф., Солодов М.В. Численные методы оптимизации. – М.: Физматлит, 2003
8. Калихман И.Л., Войтенко М.А. Динамическое программирование в примерах и задачах. – М.: Высшая школа, 1979
9. Карманов В.Г. Математическое программирование. - М.: Физматлит, 2000
10. Сборник задач по математике для втузов. Часть 4. Методы оптимизации. Уравнения в частных производных. Интегральные уравнения. – М.:Наука, 1990
11. Оптимизация функций и динамических процессов. / Сост. Городецкий С.Ю., Павлюченок З.Г., Савельев В.П. – Н.Новгород: ННГУ, 2001
12. Стронгин Р.Г. Численные методы в многоэкстремальных задачах (информационно-статистические алгоритмы). – М.: Наука, 1978
13. Чистяков В. П. Курс теории вероятностей. — 3-е изд. — М.: Наука, 1987
14. Чжун К. Однородные цепи Маркова . – М.: «Мир», 1964

15. Ширяев А. Н. Вероятность. — 2-е изд. — М.: Наука, 1989

Электронные ресурсы

16. Корпоративный портал ТГУ [Электронный ресурс]. – Режим доступа: http://portal.tltsu.ru/Информационный сайт с публикациями email сервисов [Электронный ресурс]. – Режим доступа: https://habrahabr.ru

Литература на иностранном языке

17. Nicolas Privault. Understanding Marcov Chains. Examples and Applications // Springer, 2018

18. George W. Cobb. What is Markov Chain Monte Carlo and Why it Matters // MCMC, 2018

19. Andrew Metcalfe, David Green, Tony Greenfield, Mayhayaudin Mansor. Statistics in Engineering // CRC Press, 2019

20. Xin-She Yang, Xing-Shi He. Mathematical Foundations of Nature-Inspired Algorithms // Springer, 2019

21. Rick Durrett. Probability. Theory and Examples // Cambridge Series, 2019