

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи

ЦВЕТКОВА ИННА ВЛАДИМИРОВНА

**МОДЕЛИРОВАНИЕ ХААРОВСКИХ РАСШИРЕНИЙ
СТАТИЧЕСКИХ ПРОЦЕССОВ С ПОМОЩЬЮ
ИНТЕРПОЛЯЦИОННЫХ МАРТИНГАЛЬНЫХ МЕР**

05.13.18 — Математическое моделирование, численные методы и комплексы программ

ДИССЕРТАЦИЯ

на соискание ученой степени
кандидата физико-математических наук

Научный руководитель — доктор физико-математических наук, профессор
Павлов И.В.

Ростов-на-Дону
2017

Оглавление

СПИСОК СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	4
ВВЕДЕНИЕ	6
1 Специальные хааровские интерполяции безарбитражных финансовых рынков, определённых на счётном вероятностном пространстве	27
1.1 (\mathbf{B}, \mathbf{S}) -рынок со счётным числом состояний. Основные обозначения, определения и факты.	28
1.2 Исследование (\mathbf{B}, \mathbf{S}) -рынков с помощью аргументов двойственности	38
1.3 Общий алгоритм проведения финансовых расчётов на одношаговых (\mathbf{B}, \mathbf{S}) -рынках со счётным числом состояний . .	46
2 Условия существования мартингалльных мер, удовлетворяющих ослабленному свойству универсальной хааровской единственности, в случае счётного вероятностного пространства	51
2.1 Выполнение ОУНБ для различных моделей одношаговых рынков с бесконечным числом состояний	52
2.2 ОУНБ в случае, когда последовательность $\{b_i\}_{i=1}^{\infty}$ содержит r различных значений ($3 \leq r < \infty$)	55

2.3	Построение мартингальных мер, удовлетворяющих ОУНБ в случае, когда последовательность $\{b_i\}_{i=1}^{\infty}$ содержит 3 различных значения	67
2.4	Множество мартингальных мер , удовлетворяющих ОУНБ в случае, когда последовательность $\{b_i\}_{i=1}^{\infty}$ содержит бесконечное число различных значений	74
3	Программная реализация вычисления компонент хеджирующего портфеля в случае счётного вероятностного пространства и конечнозначных цен акций	78
3.1	Описание основных алгоритмов	79
3.2	Архитектура программного комплекса	91
3.3	Результаты расчетов на примерах	103
	ЗАКЛЮЧЕНИЕ	116
	ЛИТЕРАТУРА	118
	ПРИЛОЖЕНИЕ	134

СПИСОК СОКРАЩЕНИЙ И АББРЕВИАТУР

с.в. — случайная величина;

с.п. — случайный процесс;

п.н. — почти наверное;

ф.о. — финансовое обязательство;

с.х.и.ф. — специальная хааровская интерполирующая фильтрация;

СУХЕ — свойство универсальной хааровской единственности;

ОСУХЕ — ослабленное свойство универсальной хааровской единственности;

ОУНБ — ослабленное условие несовпадения барицентров.

СПИСОК ОБОЗНАЧЕНИЙ

$\mathbb{N} = \{1, 2, 3, \dots\}$ — множество натуральных чисел;

\mathbb{R} — множество действительных чисел;

(Ω, \mathcal{F}) — измеримое пространство с множеством исходов Ω и σ -алгеброй \mathcal{F} ;

(Ω, \mathcal{F}, P) — вероятностное пространство;

$E[f|\mathcal{G}] = E^P[f|\mathcal{G}]$ — условное математическое ожидание с.в. f по вероятностной мере P относительно σ -подалгебры \mathcal{G} ;

\mathcal{P} — множество невырожденных вероятностных мер на (Ω, \mathcal{F}) , нагружающих все атомы σ -алгебры \mathcal{F} ;

$\mathbf{F} = (\mathcal{F}_k)_{k=0}^N$ — фильтрация с конечным ($N < \infty$) горизонтом;

(Ω, \mathbf{F}) — фильтрованное пространство;

$\mathbf{B} = (B_k)_{k=0}^N$ — детерминированная последовательность положительных чисел (банковский счет);

$\mathbf{S} = (S_k, \mathcal{F}_k)_{k=0}^N$ — \mathbf{F} -адаптированная последовательность случайных величин (эволюция цены акции определенного типа);

$Z = (Z_k, \mathcal{F}_k)_{k=0}^N$ — \mathbf{F} -адаптированная последовательность случайных величин (дисконтированная стоимость акции);

$\mathcal{P}(Z, \mathbf{F})$ — множество невырожденных вероятностных мер $P \in \mathcal{P}$ таких, что процесс $Z = (Z_k, \mathcal{F}_k, P)_{k=0}^N$ является мартингалом;

$\overline{\mathcal{P}}(Z, \mathbf{F})$ — множество **всех** вероятностных мер P на (Ω, \mathcal{F}) , для которых процесс $Z = (Z_k, \mathcal{F}_k, P)_{k=0}^N$ является мартингалом;

$\text{ОУНБ}(Z)$ — множество мартингальных мер, удовлетворяющих ОУНБ;

B_i — атом, получающийся в результате дробления Ω , $i \in \mathbb{N}$;

$\mathbf{H} = (\mathcal{H}_n)_{n=0}^\infty$ — хааровская фильтрация ($\mathcal{H}_0 = \{\Omega, \emptyset\}$, $\mathcal{H}_\infty = \mathcal{F}_N$), характеризующаяся следующим образом: каждая σ -алгебра \mathcal{H}_n порождена разбиением Ω ровно на $n + 1$ атом;

$Y = (Y_n, \mathcal{H}_n)_{n=0}^\infty$ — \mathbf{H} -адаптированная последовательность с.в.;

$\pi = (\beta_k, \gamma_k)_{k=0}^N$ — самофинансируемый портфель ценных бумаг (финансовая стратегия);

β_k — \mathcal{F}_{k-1} -измеримая случайная величина, выражающая количество единиц банковского счета в портфеле в момент времени $k = 0, 1, \dots, N$;

γ_k — \mathcal{F}_{k-1} -измеримая случайная величина, выражающая количество акций в портфеле в момент времени $k = 0, 1, \dots, N$;

X_k^π — полный капитал портфеля π в момент времени $k = 0, 1, \dots, N$;

F_N — \mathcal{F}_N -измеримая случайная величина (платежное обязательство);

K — цена поставки;

C — договорная цена контракта (премия);

C_N — справедливая цена финансового обязательства F_N ;

C^* — верхняя цена платежного обязательства F_N ;

P^* — мартингальная мера, соответствующая договорной цене C^* ;

C_* — нижняя цена финансового обязательства F_N ;

$|M|$ — число элементов множества M .

ВВЕДЕНИЕ

Актуальность темы. В связи с высокой динамикой событий, происходящих на финансовых рынках, возникает потребность в создании инструментов, позволяющих производить сложные финансовые расчёты. Создание таких инструментов тесно связано с разработкой соответствующих алгоритмов и программных комплексов.

В настоящее время основные исследования в финансовой математике связаны с так называемыми неполными финансовыми рынками. В диссертации рассматриваются неполные статические (\mathbf{B}, \mathbf{S}) -рынки со счётным числом состояний. Основная задача, решаемая в диссертации — расширение таких рынков до полных. Используемые в работе методы и техника расчётов основаны на теории хааровских интерполяций финансовых рынков, предложенной проф. И.В. Павловым и развитой им и его учениками М.Н. Богачёвой, А.Г. Данекянц, Т.А. Волосатовой, Г.А. Можяевым, Э.А. Пилосян. В настоящей диссертации эта техника получила своё дальнейшее развитие. Исследования основаны на изучении важного интерполяционного свойства мартингальных мер — ослабленного свойства универсальной хааровской единственности (ОСУХЕ). С помощью мартингальных мер, удовлетворяющих этому свойству и с применением специальных хааровских интерполяций исходной одношаговой фильтрации можно интерполировать заданные неполные финансовые рынки до полных.

За последние сорок лет наблюдается бурное развитие методов стохастического анализа, позволяющих моделировать эволюцию цен акций, облигаций, опционов, фьючерсов и других вторичных ценных бумаг. Многие фундаментальные факты в этом направлении были получены в результате использования теории мартингалов. В нашей стране исследование проблем финансовой математики было инициировано в начале

90-х годов академиком РАН, профессором А.Н. Ширяевым и участниками руководимых им научных семинаров в Математическом институте им. В.А. Стеклова РАН и Московском государственном университете им. М.В. Ломоносова. В список этих ученых входят А.А. Гуцин, Ю.М. Кабанов, Д.О. Крамков, А.В. Мельников, А.А. Новиков и многие другие. На юге России в указанном направлении активно работают И.В. Павлов, Д.Б. Рохлин и их ученики. Среди иностранных учёных следует отметить работы Ф.Делбаена, Ж. Жакода, Д. Зондермана, М. Йора, Х. Фельмера, В. Шахермайера, А. Шида, С. Шрива и многих других. Однако, в разработке и применении интерполяционных методов, создании соответствующих алгоритмов и комплексов программ южно-российская школа находится на передовых позициях.

Задача преобразования неполных рынков в полные впервые была рассмотрена в 1987 году в работе М.Такку и В.Виллингера [79], где переход от неполных рынков к полным осуществлялся заменой исходной мартингальной меры неэквивалентной ей мартингальной мерой. Иная методика была предложена А.В. Мельниковым и К.М. Феоктистовым в 2001 году [40]. В их работе пополнение финансового рынка проводилось посредством добавления к акциям исходного рынка дополнительных рисков активов, функционально зависящих с изначальными. В работах И.В.Павлова, М.Н.Богачёвой [10], [11] была заложена основа принципиально иного метода перехода от неполных рынков к полным на конечных вероятностных пространствах. Для решения этой задачи был использован метод интерполяций финансовых рынков, связанный с применением хааровских фильтраций и интерполяций мартингалов. Основными техническими инструментами здесь являлись мартингальные меры, удовлетворяющие свойству универсальной хааровской единственности (СУХЕ). Понятие ОСУХЕ на конечном вероятностном пространстве не имеет су-

ществленных отличий от понятия СУХЕ. Другая картина возникает на счётном вероятностном пространстве. В этом случае до сих пор не построен пример мартингальной меры, удовлетворяющей СУХЕ. Понятие ОСУХЕ было впервые введено в работе [54]. Однако, в то время была получена лишь одна модель финансового рынка, на которых существовали мартингальные меры, удовлетворяющие ОСУХЕ — это так называемый бесконечномерный вариант безарбитражной неполной модели Кокса-Росса-Рубинштейна [25]. Автором настоящей работы получен целый спектр таких моделей.

Всё вышесказанное позволяет считать тематику данной диссертации весьма актуальной.

Объектами исследования настоящей диссертации являются процессы (имитирующие эволюцию цен акций), которые допускают мартингальные меры, обладающие ОСУХЕ.

Целью диссертационной работы является изучение ОСУХЕ мартингальных мер, применение данного свойства для интерполяции мартингалов и статических финансовых рынков со счётным числом состояний, реализация построенной теории в виде вычислительных методов и алгоритмов, внедрённых в соответствующий программный комплекс. **Для реализации этой цели** потребовалось решить следующие задачи:

- 1) доказать теоремы существования мартингальных мер, удовлетворяющих ОСУХЕ;
- 2) построить алгоритмы для вычисления мартингальных мер, удовлетворяющих ОСУХЕ;
- 3) разработать алгоритмы вычислений цен акций на статическом (\mathbf{B}, \mathbf{S}) -рынке с бесконечным горизонтом, интерполирующих цены акций на исходном статическом (\mathbf{B}, \mathbf{S}) -рынке с бесконечным числом

состояний,

- 4) получить формулы хеджирования произвольных финансовых обязательств на статических (\mathbf{B}, \mathbf{S}) -рынках со счётным числом состояний с применением мартингалльных мер, удовлетворяющих ОСУХЕ;
- 5) разработать алгоритмы вычислений цен акций на рынке, интерполирующем исходный рынок, а также алгоритмы вычисления справедливых цен финансовых обязательств и компонент хеджирующих портфелей;
- 6) создать программный комплекс, реализующий полученные алгоритмы и формулы.

Методика исследований. При решении перечисленных задач применялись методы и результаты теории вероятностей, теории мартингалов и мартингалльных мер, теории чисел (p -адические представления), методы решения оптимизационных задач линейного программирования с бесконечным числом переменных, теории алгоритмов и структур данных, имитационное моделирование.

Разработка программного комплекса велась в среде Qt Creator, фреймворк Qt5. В качестве основного алгоритмического языка выбран объектно-ориентированный язык C++. Дополнительно использовались: открытая «библиотека» функций для решения задач линейного программирования (GLPK – GNU Linear Programming Kit); модифицированный парсер и визуализатор языка MathML из открытой «библиотеки» Qwt. Программный комплекс состоит из основной программы (загрузчика модулей) и подгружаемых модулей (представленных в виде динамических библиотек — DLL, реализующих графические интерфейсы и расчетную часть). Организован по принципу фреймворка таким образом, что предоставляет программисту, пишущему модуль, базовый набор примитивов

для организации работы с деревьями событий, их отображением, расчетами на них, а также позволяет реализовать интуитивно понятный пользовательский интерфейс. Такая логическая организация позволяет развивать и дополнять программный комплекс новыми моделями. Представленный программный комплекс отвечает следующим требованиям: поддерживает все системы семейства Windows, начиная с Windows XP (т.е. Windows XP/Vista/7/8/8.1/10), высокая производительность, эргономичный пользовательский интерфейс.

Научная новизна и выносимые на защиту результаты. Все основные результаты диссертации являются новыми. Перечислим наиболее важные из них, которые выносятся на защиту:

1. Доказательство существования мартингалльных мер, удовлетворяющих ОСУХЕ, в случае произвольного начального значения цены акции и **трёх** произвольных значений (всевозможных кратностей) финальной цены акции. Создание алгоритма для вычисления этих мартингалльных мер.
2. Доказательство существования мартингалльных мер, удовлетворяющих ОСУХЕ, в случае произвольного начального значения цены акции и **любого конечного числа** рациональных значений (всевозможных кратностей) финальной цены акции. Создание алгоритма для вычисления полученных мартингалльных мер.
3. Доказательство существования мартингалльных мер, удовлетворяющих ОСУХЕ, когда начальное значение цены акции находится в определённом интервале, а значения цены акции в финальный момент времени экспоненциально растут.
4. Новые формулы для вычисления совершенных хеджирующих стратегий на интерполирующем рынке (\mathbf{B}, \mathbf{S}) -рынке с бесконечным го-

ризонтом с использованием мартингальных мер, удовлетворяющих ОСУХЕ.

5. Разработанные приближённые методы квантильного хеджирования на интерполирующих (\mathbf{B}, \mathbf{S}) -рынках.
6. Новый научно-исследовательский программный комплекс, написанный на языке C++ в среде Qt Creator, фреймворк Qt5, реализующий полученные в диссертации алгоритмы и формулы.

Теоретическая и практическая ценность. Работа носит как теоретический, так и прикладной характер. Результаты диссертации могут использовать хеджеры и эмитенты акций и вторичных ценных бумаг, когда на рынке происходит массовая скупка рискованных активов. Полученные в диссертации теоретические результаты, связанные с достаточными условиями существования мартингальных мер, удовлетворяющих ОСУХЕ, вносят значительный вклад в стохастическую финансовую математику. Основные результаты работы, а также алгоритмы программного комплекса и непосредственно программный комплекс могут найти применение в ряде компаний, использующих современные компьютерные технологии.

Область исследования. В диссертации рассматриваются задачи, связанные с развитием теории интерполяции неполных безарбитражных финансовых рынков. Полученные теоретические результаты вносят вклад в стохастическую финансовую математику.

Достоверность результатов работы подтверждается:

- 1) математическими доказательствами, результатами моделирования и обработки данных;
- 2) апробацией этих результатов на всероссийских и международных

конференциях и научных семинарах.

Апробация работы. Основные результаты работы диссертации докладывались на следующих международных, всероссийских и региональных научных конференциях и научных семинарах:

- 1) на Международных конференциях «Современные методы и проблемы теории операторов и гармонического анализа и их приложения» (г. Ростов-на-Дону, 2011-2016 гг.);
- 2) на Международной научно-практической конференции «Теория операторов, комплексный анализ и математическое моделирование», секция «Дифференциальные уравнения и математическое моделирование» (г. Волгодонск, 2011 г.);
- 3) на Международных научно-практических конференциях «Строительство», секция «Математика» (г. Ростов-на-Дону, 2011-2016 гг.);
- 4) на XX Международной конференции «Математика. Экономика. Образование» (г. Ростов-на-Дону, 2012 г.);
- 5) на VI Международной конференции «ПМТУКТ-2013» «Современные методы прикладной математики, теории управления и компьютерных технологий» (г. Воронеж, 2013 г.);
- 6) на XIV Всероссийском симпозиуме по прикладной и промышленной математике (г. Великий Новгород, 2013 г.);
- 7) на Международной конференции «XXV Крымская Осенняя Математическая Школа-симпозиум по спектральным и эволюционным задачам», секция «Теория вероятностей. Случайные процессы. Финансовая математика. Математическая статистика» (г. Судак, 2014 г.);

- 8) на Международной конференции «XXVI Крымская Осенняя Математическая Школа-симпозиум по спектральным и эволюционным задачам», секция «Теория вероятностей. Случайные процессы. Финансовая математика. Математическая статистика.» (г. Симферополь, 2015 г.);
- 9) на Международной конференции по стохастическим методам (пос. Абрау-Дюрсо, 2016 г.);
- 10) на научном семинаре по финансовой математике и стохастическому моделированию при кафедре "Высшая математика" ДГТУ (рук. — д.ф.-м.н., проф. Павлов И.В.).

Часть исследований, результаты которых представлены в диссертации, поддержаны грантами РФФИ (проекты 13-01-00637а и 16-01-00184).

Публикации. По теме диссертации опубликовано 22 печатные работы, 1 свидетельство о регистрации программы [104]. Работы [89], [91], [94], [99] опубликованы в журналах из перечня рецензируемых научных изданий, рекомендованных ВАК Минобрнауки РФ, работы [100], [102] опубликованы в журнале «Теория вероятностей и её применения», индексируемом в базе данных Web of Science. Из совместных работ [82]-[97],[100] в диссертацию вошли результаты, полученные лично диссертантом.

В работах [82],[84],[89] автором была построена концепция решения бесконечномерной задачи оптимизации, возникающей при исследовании финансовых рынков со счётным числом состояний. В работах [83],[86],[87],[88],[95],[96],[97],[100] для различных моделей одношаговых (\mathbf{B}, \mathbf{S}) -рынков автором были получены достаточные условия существования мартингалльных мер, удовлетворяющих ослабленному условию несовпадения барицентров (ОУНБ). Была проведена классифика-

ция множеств мартингалльных мер, удовлетворяющих ОУНБ, в зависимости от значений дисконтированной стоимости акций. В работах [90],[91],[93],[94] автором были разработаны вычислительные схемы построения совершенных хеджей для ограниченных финансовых обязательств F_1 . В работе [92] исследована модель случайного поведения скупщиков в промежуточные моменты времени между объявлениями цен акций. В работах [98],[99],[101],[102],[103] автором были разработаны алгоритмы построения интерполяционных мартингалльных мер в случае счётного вероятностного пространства и конечнозначных цен акций, а также алгоритмы вычислительных процедур при построении хеджирующих стратегий на финансовых рынках с бесконечным числом состояний.

Структура и объём диссертации. Диссертация состоит из введения, 3-х глав, заключения, списка литературы (104 наименования), приложения. Каждая глава разбита на параграфы. Нумерация параграфов двойная: первая цифра указывает номер главы, в которой расположен параграф, а вторая цифра — номер самого параграфа. Такая же нумерация определений, теорем, лемм т.п. Нумерация рисунков сплошная. Работа проиллюстрирована 35 рисунками и изложена на 133 страницах (без приложения).

Краткое изложение основных результатов диссертации.

Обзор результатов 1-ой главы. Мы рассматриваем (\mathbf{B}, \mathbf{S}) -рынок, который определяется безрисковым активом $\mathbf{B} = (B_k)_{k=0}^N$, понимаемый нами как размер банковского счёта в момент времени k , и рисковым активом $\mathbf{S} = (S_k)_{k=0}^N$, где $S_k = (S_k^{(1)}, S_k^{(2)}, \dots, S_k^{(l)})$ — вектор цен акций в момент времени k . N — финальный момент времени, до которого исследуется рассматриваемая модель ($N < \infty$). Определим фильтрованное пространство $(\Omega, \mathcal{F}, \mathbf{F}, P)$, в котором:

Ω — счётное пространство элементарных событий ;

\mathcal{F} — σ -алгебра подмножеств пространства Ω ;

\mathbf{F} — фильтрация, состоящая из возрастающей последовательности σ -алгебр:

$$\mathcal{F}_0 \subset \mathcal{F}_1 \subset \dots \subset \mathcal{F}_N, \quad (\mathcal{F}_0 = \{\Omega, \emptyset\}, \mathcal{F}_N = \mathcal{F})$$

P — вероятностная мера, нагружающая все атомы Ω .

При этом каждый элемент стохастической последовательности $\mathbf{S} = (S_k)_{k=0}^N$ является \mathcal{F}_k -измеримой с.в., а $\mathbf{B} = (B_k)_{k=0}^N$ — детерминированная последовательность положительных чисел. Пусть β_k — количество единиц банковского счёта, а γ_k — количество акций в момент времени k (с.в. β_k, γ_k являются \mathcal{F}_{k-1} -измеримыми, $1 \leq k \leq N$). Стохастическая последовательность $\pi = (\beta_k, \gamma_k)_{k=0}^N$ называется портфелем ценных бумаг (инвестиционной стратегией). Полным капиталом портфеля π в момент времени k , $1 \leq k \leq N$ называется случайная величина:

$$X_k^\pi = \beta_k B_k + \gamma_k S_k.$$

В промежутке между моментами времени $k-1$ и k ($1 \leq k \leq N$) в результате различных обстоятельств капитал X_{k-1}^π может измениться и принять значение:

$$X_{k-1}^\pi + h_k = \beta_k B_{k-1} + \gamma_k S_{k-1}$$

(h_k — \mathcal{F}_{k-1} измеримая с.в.). Но в момент времени k после объявления новых цен на акции и учитывая изменения стоимости банковского счёта, получим:

$$X_k^\pi = \beta_k B_k + \gamma_k S_k.$$

Портфель π , обладающий свойством:

$$B_{k-1} \Delta \beta_k + S_{k-1} \Delta \gamma_k = 0$$

называется *самофинансируемым*. Условие самофинансирования означает, что перед изменением состава портфеля π в промежутке между мо-

ментами $k - 1$ и k портфель не испытывает ни притока дополнительного капитала, ни его оттока.

Одним из основных понятий финансовой математики является безарбитражность и полнота финансовых рынков. Портфель π реализует арбитражную возможность тогда и только тогда, когда:

$$X_0^\pi = 0, P(X_k^\pi \geq 0) = 1, P(X_N^\pi > 0) > 0.$$

Иными словами, арбитражем называется возможность получения прибыли при нулевых затратах. (\mathbf{B}, \mathbf{S}) -рынок, на котором отсутствуют арбитражные возможности называется безарбитражным. В дальнейшем мы будем рассматривать только безарбитражные рынки.

Пусть F_N — \mathcal{F}_N -измеримая неотрицательная с.в. (финансовое обязательство). Самофинансируемый портфель π , который бы позволил в финальный момент времени N достигнуть некоторое финансовое обязательство F_N (то есть $X_N^\pi = F_N$) называется хеджирующим портфелем. (\mathbf{B}, \mathbf{S}) -рынок называется полным, если любое финансовое обязательство F_N реплицируемо.

Рассмотрим \mathbf{F} -адаптированный случайный процесс $Z = (Z_k, \mathcal{F}_k)_{k=0}^N$ ($Z_k = \frac{S_k}{B_k}$, $k = 0, 1, \dots, N$), который мы будем мыслить как дисконтированную стоимость акции. Цена банковского счёта дисконтированного рынка тождественно равна единице.

Пусть \mathcal{P} — множество невырожденных вероятностных мер на (Ω, \mathcal{F}) .

Определение 0.1 *Вероятностная мера $P \in \mathcal{P}$ называется мартингальной мерой, если с.п. $Z = (Z_k, \mathcal{F}_k)_{k=0}^N$ является мартингалом:*

$$E^P[Z_{k+1} | \mathcal{F}_k] = Z_k, \quad k = 0, 1, 2, \dots, N - 1$$

Обозначим через $\mathcal{P}(Z, \mathbf{F})$ — множество мартингальных мер с.п. Z . Сформулируем фундаментальные теоремы финансовой математики.

Теорема 0.1 Для того, чтобы (\mathbf{B}, \mathbf{S}) -рынок был безарбитражен, необходимо и достаточно, чтобы $\mathcal{P}(Z, \mathbf{F}) \neq \emptyset$.

Теорема 0.2 Для того, чтобы безарбитражный (\mathbf{B}, \mathbf{S}) -рынок был полон, необходимо и достаточно, чтобы множество $\mathcal{P}(Z, \mathbf{F})$ состояло из одного элемента.

Мы всё время предполагаем, что $\mathcal{P}(Z, \mathbf{F}) \neq \emptyset$, то есть исходный рынок является безарбитражным.

В диссертационной работе мы будем рассматривать статические модели (т.е. $N = 1$) (\mathbf{B}, \mathbf{S}) -рынков. Дерево стохастического базиса таких моделей изображено на рис. 1.

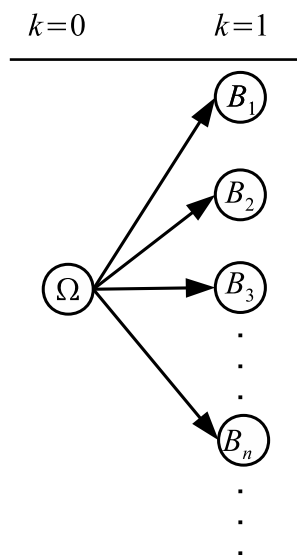


Рис. 1

Рассмотрим некоторую перестановку натуральных чисел: $\delta = (\delta^{(1)}, \delta^{(2)}, \dots)$. Координаты вектора δ — зависимые случайные величины, определяемые с помощью генератора случайных чисел, основанном на геометрическом распределении [16].

Если исходить из того, что новые события на рынке возникают неодновременно, тогда предполагая, что при переходе от момента времени $k = 0$

к моменту $k = 1$ атом Ω дробится на счётное число атомов B_1, B_2, \dots , можно определить следующие σ -алгебры событий: $\mathcal{H}_0 = \mathcal{F}_0$, $\mathcal{H}_1 = \{\Omega, \emptyset, B_{\delta^{(1)}}, \bar{B}_{\delta^{(1)}}\}$ (т.е. $\mathcal{H}_1 = \sigma\{B_{\delta^{(1)}}\}$), $\mathcal{H}_2 = \sigma\{B_{\delta^{(1)}}, B_{\delta^{(2)}}\}, \dots, \mathcal{H}_\infty = \sigma\{B_{\delta^{(1)}}, B_{\delta^{(2)}}, \dots\}$. Ясно, что $\mathcal{H}_0 = \mathcal{F}_0$, $\mathcal{H}_\infty = \mathcal{F}_1$. Поэтому полученная фильтрация $\mathbf{H} = (\mathcal{H}_n)_{n=0}^\infty$ является интерполирующей фильтрацией для исходной фильтрации $\mathbf{F} = (\mathcal{F}_k)_{k=0}^1$. Изучаемая фильтрация \mathbf{H} обладает следующим свойством: при переходе от момента времени n к моменту времени $n + 1$ только один атом из \mathcal{H}_n дробится на две части, причём тот, который в результате дробления был получен на предыдущем шаге, а остальные атомы этой σ -алгебры остаются неизменными. Такие фильтрации называются специальными хааровскими фильтрациями рис. 2.

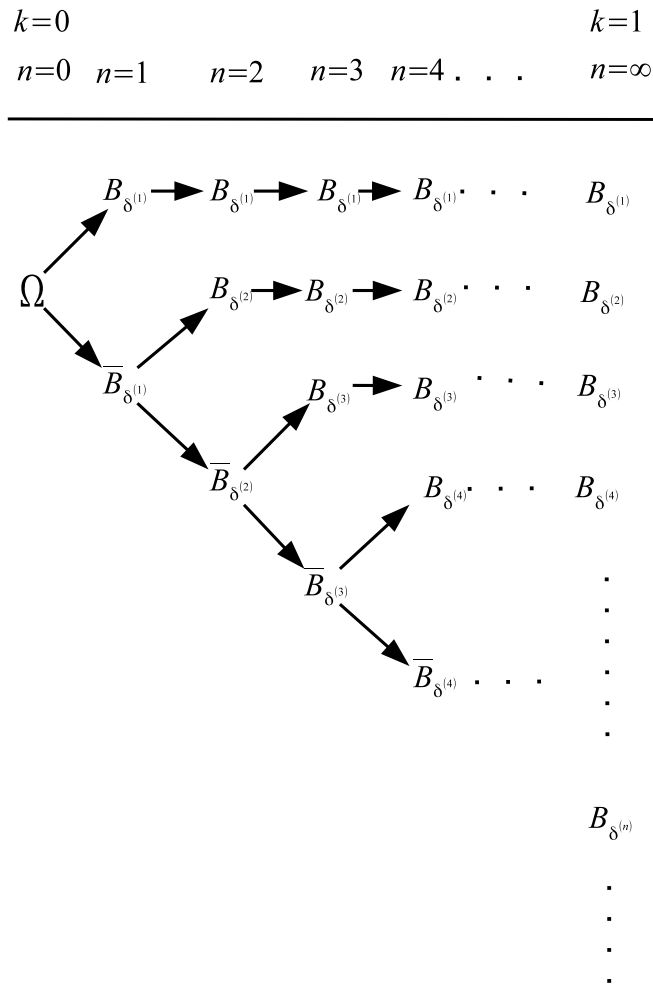


Рис. 2

Итак, пусть $\Omega = \bigcup_{i=1}^{\infty} B_i$. Обозначим значения дисконтированной цены акции:

$$a := Z_0|_{\Omega}, \quad b_i := Z_1|_{B_i}, \quad a, b_i \in \mathbb{R}, \quad i \in \mathbb{N}. \quad (0.1)$$

Зафиксируем мартингальную меру $P \in \mathcal{P}(Z, \mathbf{F})$. Для мартингала $Z = (Z_k, \mathcal{F}_k, P)_{k=0}^1$ по формуле $Y_n = E^P[Z_1 | \mathcal{H}_n]$ построим мартингальную хааровскую интерполяцию $Y = (Y_n, \mathcal{H}_n, P)_{n=0}^{\infty}$ (если с. х. и. ф. \mathbf{H} фильтрации \mathbf{F} фиксирована, то мартингал Y определяется по мартингалу Z однозначно).

Определение 1.8 Будем говорить, что мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетворяет ослабленному свойству универсальной хааровской единственности (ОСУХЕ), если для любой специальной хааровской интерполяции \mathbf{H} исходной фильтрации \mathbf{F} и соответствующей хааровской интерполяции $Y = (Y_n, \mathcal{H}_n)_{n=0}^{\infty}$ мартингала $Z = (Z_k, \mathcal{F}_k, P)_{k=0}^N$ имеет место равенство $|\mathcal{P}(Y, \mathbf{H})| = 1$.

Определение 1.9 Будем говорить, что мартингальная мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетворяет ослабленному условию несовпадения барицентров (ОУНБ), если:

- 1) ряд $\sum_{i=1}^{\infty} b_i p_i$ сходится абсолютно;
- 2) $\forall i = 1, 2, \dots$ и для любого набора индексов $J \subset \{1, 2, \dots\}$ такого, что $i \notin J$; \bar{J} — конечно, выполняется неравенство:

$$b_i \neq \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j}. \quad (0.2)$$

Для перехода от неполных безарбитражных рынков к полным необходимо, чтобы рассматриваемая модель обладала мартингальными мерами, удовлетворяющими ОСУХЕ. Однако это свойство является неудобным с точки зрения аналитической проверки его выполнения. В теоре-

ме 1.2 доказано, что ОСУХЕ равносильно ОУНБ.

Для того, чтобы обосновать актуальность задачи о пополнении безарбитражного рынка, коротко рассмотрим понятие опциона. Для конкретики рассмотрим опцион Европейского типа. Как известно, такие опционы дают его владельцу право купить актив по цене K , которая зафиксирована в договоре. Очевидно, что в финальный момент времени N фактическая стоимость акции (S_N) может отличаться от контрактной. В случае, если $S_N > K$, то владельцу опциона выгодно предъявить его к исполнению, так как тут же продав эти акции, он получит доход, равный $S_N - K$. В случае когда $S_N < K$, тогда опцион не предъявляется к исполнению, так как сделка становится невыгодной. Таким образом, в финальный момент времени N доход покупателя (равный финансовому обязательству продавца) определяется по формуле:

$$F_N = (S_N - K)^+ = \max\{S_N - K, 0\}.$$

Получив за проданный опцион премию C_N , продавец опциона должен к моменту N иметь возможность выполнить взятые на себя обязательства. Поэтому, имея начальный капитал X_0^π , он должен так построить свою стратегию, чтобы в момент N достигнуть платёжного обязательства F_N , совпадающего с доходом покупателя, т.е. $X_N^\pi = (S_N - K)^+$.

В связи с этим, оказываются весьма актуальными задачи о вычислении справедливой цены опциона C_N и о построении стратегии продавца опциона, хеджирующей его финансовое обязательство.

Очевидно, что продавец и покупатель опциона преследуют разные цели, назначая продажную и покупную цены опциона. Различие целей продавца и покупателя приводит к назначению, вообще говоря, разных цен продажи $C^*(N)$ и покупки $C_*(N)$. Поэтому в случае неполного рынка, это приводит к появлению ненулевой разницы $C^* - C_*$, называемой

спредом. Если же рынок полон, то возможно совмещение противоположных интересов продавца и покупателя, выражающееся в существовании справедливой цены опциона C_N :

$$C_N = C^* = C_*.$$

С нематематической точки зрения, свойство полноты обеспечивает доступность всех фигурирующих на рынке активов и отсутствие ограничений для инвестирования в эти активы. Общие формулы расчета справедливой цены и хеджирующих стратегий приводятся в рамках полного безарбитражного рынка. Поэтому вопрос о пополнении безарбитражного неполного рынка достаточно актуален.

Обзор результатов 2-ой главы диссертации. Вторая глава диссертации посвящена определению достаточных условий существования мартингалльных мер, удовлетворяющих ОУНБ, для различных одношаговых моделей (\mathbf{B}, \mathbf{S}) -рынков, определённых на счётном вероятностном пространстве.

С учётом обозначений (0.1), рассмотрим систему :

$$\begin{cases} \sum_{i=1}^{\infty} p_i = 1 \\ \sum_{i=1}^{\infty} b_i p_i = a \\ p_i > 0, i \in \mathbb{N}. \end{cases} \quad (0.3)$$

Множество решений этой системы совпадает с множеством вероятностных мер $P \in \mathcal{P}$, для которых с.п. $(Z_k, \mathcal{F}_k, P)_{k=0}^1$ является мартингаллом. Обозначим через $|M|$ — число элементов некоторого множества M . Очевидно, что $\mathcal{P}(Z, \mathbf{F})$ является выпуклым подмножеством множества \mathcal{P} . Отсюда вытекает, что существует три возможности:

- 1) $\mathcal{P}(Z, \mathbf{F}) = \emptyset$ (в этом случае финансовый рынок является арбитраж-

ным);

2) $|\mathcal{P}(Z, \mathbf{F})| = 1$ (в этом случае рынок является полным);

3) $|\mathcal{P}(Z, \mathbf{F})| = \infty$ (в этом случае рынок является неполным).

В дальнейшем мы будем предполагать, что выполняется условие:

$$\inf_{1 \leq i < \infty} b_i < a < \sup_{1 \leq i < \infty} b_i. \quad (0.4)$$

Это условие является достаточным условием того, что рассматриваемый рынок является безарбитражным и неполным.

Определение 2.1 Будем говорить, что число b , входящее в последовательность $\{b_i\}_{i=1}^{\infty}$, имеет кратность t (t может быть как конечно, так и бесконечно), если в этой последовательности оно присутствует t раз.

Обозначим через $\text{ОУНБ}(Z)$ — множество мартингалльных мер, удовлетворяющих ОУНБ.

Предложение 2.1 Если среди элементов последовательности $\{b_i\}_{i=1}^{\infty}$ конечное число значений имеет конечную кратность и одно значение имеет бесконечную кратность, тогда ни одна мера $P \in \mathcal{P}(Z, \mathbf{F})$ не удовлетворяет ОУНБ.

Предложение 2.2 Если среди элементов последовательности $\{b_i\}_{i=1}^{\infty}$ только два различных и оба они имеют бесконечную кратность, тогда любая мартингалльная мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетворяет ОУНБ.

Не нарушая общности, будем считать, что элементы последовательности $\{b_i\}_{i=1}^{\infty}$ удовлетворяют неравенству $b_1 < b_2 < \dots < b_r$, $2 < r \leq \infty$. Пусть они имеют кратности m_1, m_2, \dots, m_r соответственно. Принимая во внимание сказанное выше, предполагаем, что не менее чем два из этих чисел бесконечны. Изменив нумерацию, легко получаем следующую си-

стему обозначений:

$$\begin{aligned}
 & b_1 = b_{r+1} = b_{2r+1} = \dots = b_{r(j-1)+1} = \dots, 1 \leq j < m_1 + 1, \\
 & \dots\dots\dots \\
 & b_k = b_{r+k} = b_{2r+k} = \dots = b_{r(j-1)+k} = \dots, 1 \leq j < m_k + 1, \\
 & \dots\dots\dots \\
 & b_r = b_{2r} = b_{3r} = \dots = b_{rj} = \dots, 1 \leq j < m_r + 1.
 \end{aligned}$$

При этом система (0.3) принимает вид:

$$\begin{cases}
 \sum_{k=1}^r \sum_{j=1}^{m_k} p_{r(j-1)+k} = 1 \\
 \sum_{k=1}^r b_k \sum_{j=1}^{m_k} p_{r(j-1)+k} = a \\
 p_{r(j-1)+k} > 0, 1 \leq k \leq r, 1 \leq j < m_k + 1.
 \end{cases} \quad (0.5)$$

Решение системы (0.5) будем искать следующим образом. Обозначим через:

$$x_k = \sum_{j=1}^{m_k} p_{r(j-1)+k}, \quad (0.6)$$

рассмотрим систему

$$\begin{cases}
 \sum_{k=1}^r x_k = 1 \\
 \sum_{k=1}^r b_k x_k = a \\
 x_k > 0, 1 \leq k \leq r.
 \end{cases} \quad (0.7)$$

Получив решение (x_1, \dots, x_r) системы (0.7) и записав $\forall k(1 \leq k \leq r)$ произвольное представление (0.6), получим решение системы (0.5).

Теорема 2.1 Пусть среди элементов последовательности $\{b_i\}_{i=1}^{\infty}$ три различных значения: b_1, b_2, b_3 . Тогда если $b_1 < a < b_2 < b_3$, $m_1 = 1, m_2 = m_3 = \infty$, (или $b_1 < b_2 < a < b_3$, $m_1 = m_2 = \infty, m_3 = 1$), то $ОУНБ(Z) = \mathcal{P}(Z, \mathbf{F})$.

Таким образом, (\mathbf{B}, \mathbf{S}) -рынок, удовлетворяющий условиям этой теоремы является полным.

Лемма 2.1 Пусть $b_1 < b_2 < \dots < b_r$, $b_1 < a < b_r$, $a \neq b_2, \dots, a \neq b_{r-1}$. Пусть l и n ($1 \leq l < n \leq r$) — два произвольных индекса. Тогда если числа b_1, \dots, b_r и a рациональны и $b_l < a < b_n$, то существуют решения системы (0.7) (x_1, x_2, \dots, x_r) следующих двух типов:

- 1) все компоненты решения рациональны, а числа x_k ($k \neq l, n$) сколь угодно малы;
- 2) компоненты x_l, x_n и x_s иррациональны (s — произвольный фиксированный индекс, $s \neq l, n$), остальные компоненты решения рациональны, числа x_k ($k \neq l, n$) сколь угодно малы.

Используя результаты леммы 2.1, можно доказать следующую теорему.

Теорема 2.3. Пусть числа $b_1 < b_2 < \dots < b_r$ рациональны, a — действительное число, $b_1 < a < b_r$, $a \neq b_k$ ($k = 2, 3, \dots, r-1$). Тогда $\text{ОУНБ}(Z) \neq \emptyset$.

В параграфе 2.3 представлен конструктивистский способ построения мартингалльных мер, удовлетворяющих ОУНБ, когда число различных значений среди элементов последовательности $\{b_i\}_{i=1}^{\infty}$ равно трём (всевозможных кратностей).

Для того, чтобы получить достаточное условие существования мартингалльных мер, удовлетворяющих ОУНБ в случае, когда последовательность $\{b_i\}_{i=1}^{\infty}$ содержит бесконечное число различных значений, понадобится следующая лемма.

Лемма 2.4 Пусть $P \in \mathcal{P}(Z, \mathbf{F})$ — произвольная мартингалльная мера и $b_1 < a < b_2 < b_3 < b_4 < \dots$, причём $b_i \geq 2b_{i-1}$, $\forall i \geq 2$. Для того, чтобы мера P удовлетворяла ОУНБ достаточно, чтобы

$$b_{i-1} \min_{2 \leq j \leq i-1} p_j > \sum_{j=i+1}^{\infty} b_j p_j, \forall i \geq 3. \quad (0.8)$$

В частности, если последовательность p_2, p_3, \dots монотонно убывает,

то неравенство (0.8) совпадает с неравенством

$$b_{i-1}p_{i-1} > \sum_{j=i+1}^{\infty} b_j p_j, \quad \forall i \geq 3. \quad (0.9)$$

Теорема 2.4 Пусть элементы последовательности $\{b_i\}_{i=1}^{\infty}$ удовлетворяют неравенству: $b_1 < a < b_2 < b_3 < b_4 < \dots$, причём

$$b_i \geq 2b_{i-1}, \quad \forall i \geq 2. \quad (0.10)$$

Тогда $OУНБ(Z) \neq \emptyset$ и строго вложено в $\mathcal{P}(Z, \mathbf{F})$.

В **3-ей главе** детально описан программный комплекс "Построение хеджирующих стратегий посредством хааровских интерполяций" ([104]). Разработка программного комплекса велась в среде Qt Creator, фреймворк Qt5. В качестве основного алгоритмического языка выбран объектно-ориентированный язык C++. Комплекс состоит из основной программы (загрузчика модулей) и подгружаемых модулей, представленных в виде динамически подключаемых библиотек (Dynamic Link Libraries). Такая логическая организация позволяет развивать и дополнять программный комплекс новыми моделями и вычислительными процедурами.

В **параграфе 3.1** представлено описание вычислительных процессов, которые необходимы для разработки программного комплекса. Показаны следующие алгоритмы:

- 1) алгоритм конструирования моделей одношаговых (\mathbf{B}, \mathbf{S}) -рынков с бесконечным числом состояний;
- 2) алгоритмы вычисления компонент мартингалльных мер;
- 3) алгоритм проведения финансовых расчётов;

- 4) ряд прочих алгоритмов, необходимых для реализации программного комплекса.

В *параграфе 3.2* изложена архитектура программного комплекса. Описаны принципы взаимодействия его компонент. Рассматриваются всевозможные информационные структуры, классы, диалоги.

В **заключении** приводятся основные результаты работы, выносимые на защиту.

В **приложении** представлена структура исходного кода программного комплекса.

Глава 1

Специальные хааровские интерполяции безарбитражных финансовых рынков, определённых на счётном вероятностном пространстве

Первая глава состоит из трёх параграфов. В *параграфе 1.1* вводятся основные обозначения, определения и теоремы для (\mathbf{B}, \mathbf{S}) -рынка со счётным числом состояний. Изложены некоторые факты из теории хааровских интерполяций финансовых рынков. В *параграфе 1.2* исследуется прямая и двойственная задачи для определения верхней и нижней цены финансового обязательства. В *параграфе 2.3* представлен алгоритм проведения финансовых расчётов на безарбитражных неполных (\mathbf{B}, \mathbf{S}) -рынках со счётным числом состояний.

1.1 (B, S)-рынок со счётным числом состояний. Основные обозначения, определения и факты.

В стохастической финансовой математике финансовые рынки называются (B, S)-рынками. Для того, чтобы создать модель финансового рынка, необходимо ввести в рассмотрение следующие активы:

- 1) безрисковый актив $\mathbf{B} = (B_k)_{k=0}^N$, где B_k — это детерминированная последовательность положительных чисел, представляющая цену банковского счета в момент времени k . Например, B_k может изменяться по формуле сложных процентов $B_k = B_0(1 + r)^k$, где r — банковская процентная ставка;
- 2) вектор цен акций $\mathbf{S} = (S_k^{(1)}, S_k^{(2)}, \dots, S_k^{(l)})_{k=0}^N$, где $S_k^{(i)}$ — строго положительная с.в., выражающая цену акции i -го типа в момент времени k .

Поскольку $S_k^{(i)}$ — с.в., то она должны быть определены на некотором измеримом пространстве (Ω, \mathcal{F}) . В нашей работе Ω — счётное множество, а \mathcal{F} — совокупность всех подмножеств множества Ω , представляющая собой σ -алгебру событий на Ω .

Рассмотрим возрастающую последовательность σ -алгебр:

$$\mathcal{F}_0 \subset \mathcal{F}_1 \subset \dots \subset \mathcal{F}_N,$$

где N — финальный момент времени (горизонт), до которого включительно исследуется модель финансового рынка ($N < \infty$). Такой поток σ -алгебр называют фильтрацией. Обозначим через: $\mathbf{F} = (\mathcal{F}_k)_{k=0}^N$, при этом $\mathcal{F}_0 = \{\Omega, \emptyset\}$, $\mathcal{F}_N = \mathcal{F}$. Каждую σ -алгебру \mathcal{F}_k можно трактовать как совокупную информацию о рынке в этот момент времени. Принято считать, что в момент времени $k = 0$ вся ситуация на рынке из-

вестна, то есть цена S_0^i акции i -го типа ($i = 1, 2, \dots, l$) является константой. Поэтому вектор $S_0 = (S_0^{(1)}, S_0^{(2)}, \dots, S_0^{(l)})$ неслучаен и измерим относительно тривиальной σ -алгебры \mathcal{F}_0 . Соответственно, если рассмотреть момент времени $k = 1, 2, \dots, N$, то в силу того, что все события σ -алгебры \mathcal{F}_k считаются нам известными, поэтому известен и вектор цен акций $S_k = (S_k^{(1)}, S_k^{(2)}, \dots, S_k^{(l)})$. Математически этот факт означает, что случайный вектор S_k измерим относительно \mathcal{F}_k . В этом случае говорят, что последовательность цен $(S_k)_{k=0}^N$ адаптирована к фильтрации $\mathbf{F} = (\mathcal{F}_k)_{k=0}^N$.

Обозначим через \mathcal{P} множество невырожденных вероятностных мер на (Ω, \mathcal{F}) , каждая из которых нагружает все атомы $B_i \in \Omega$, $i \in \mathbb{N}$. В дальнейшем мы будем отождествлять множество мер $P \in \mathcal{P}$ с множеством векторов $P = (p_1, p_2, \dots)$, где $p_i = P(B_i) > 0$, $i = 1, 2, \dots$, $\sum_{i=1}^{\infty} p_i = 1$.

Определение 1.1 *Стохастическим базисом $(\Omega, \mathbf{F}) = ((\mathcal{F}_k)_{k=0}^N, \mathcal{F}, P)$ называется вероятностное пространство (Ω, \mathcal{F}, P) вместе с фильтрацией $\mathbf{F} = (\mathcal{F}_k)_{k=0}^N$.*

Определение 1.2 *Рынок $\mathbf{Z} = (1, Z)$, $Z = (Z_0, Z_1, Z_2, \dots, Z_N)$, где $Z_k = \frac{S_k}{B_k}$ ($k = 0, 1, 2, \dots, N$), называется дисконтированием исходного (\mathbf{B}, \mathbf{S}) -рынка (соответственно, Z_k называется дисконтированием акции k -го типа).*

Таким образом, $Z = (Z_k, \mathcal{F}_k)_{k=0}^N$ — это \mathbf{F} -адаптированный с.п.

Определение 1.3 *Портфелем (в обобщённом смысле) ценных бумаг (или **финансовой стратегией** в обобщённом смысле) называется объект $\pi = (\beta_n, \gamma_n)_{n=0}^N$, состоящий из последовательности с.в. $\beta_0, \beta_1, \dots, \beta_N$, где каждое β_k интерпретируется как число единиц банковского счёта в момент времени k , и последовательность случайных векторов $\gamma_0, \gamma_1, \dots, \gamma_N$, где каждый случайный вектор $\gamma_k =$*

$(\gamma_k^{(1)}, \gamma_k^{(2)}, \dots, \gamma_k^{(l)})$ интерпретируется как число акций в момент времени $k = 0, 1, 2, \dots, N$.

Если считать, что с.в. β_k и случайные векторы γ_k являются \mathcal{F}_{k-1} -измеримыми, $k = 0, 1, 2, \dots, N$ ($\mathcal{F}_{-1} := \mathcal{F}_0$), то в определении 1.3 получим **портфель ценных бумаг в обычном финансовом смысле**. В дальнейшем мы именно это и предполагаем. С.в. $\beta_k, \gamma_k^i, k = 0, 1, 2, \dots, N, i = 1, 2, \dots, l$ принимают положительные, равные нулю, отрицательные значения, а также могут быть дробными.

Полным капиталом портфеля π в момент времени k ($k = 0, 1, 2, \dots, N$) называется с.в.:

$$X_k^\pi = \beta_k B_k + \gamma_k S_k, \quad (1.1)$$

где второе слагаемое понимается как скалярное произведение векторов γ_k и S_k .

Справедлива следующая теорема.

Теорема 1.1 *Рассмотрим портфель $\pi = (\beta_k, \gamma_k)_{k=0}^N$ ценных бумаг с капиталом (1.1). Тогда следующие условия равносильны:*

(a) $X_{k-1}^\pi + h_k = \beta_k B_{k-1} + \gamma_k S_{k-1}, k = 1, 2, \dots, N$ (вид финансирования портфеля);

(b) $B_{k-1} \Delta \beta_k + S_{k-1} \Delta \gamma_k = h_k, k = 1, 2, \dots, N$ (балансовое соотношение);

(c) $\Delta X_k^\pi = \beta_k \Delta B_k + \gamma_k \Delta S_k + h_k, k = 1, 2, \dots, N$ (формула приращения капитала);

(d) $\Delta \left(\frac{X_k^\pi}{B_k} \right) = \gamma_k \Delta \left(\frac{S_k}{B_k} \right) + \frac{h_k}{B_{k-1}}, k = 1, 2, \dots, N$ (формула приращения дисконтированного капитала).

Доказательство теоремы основано на результатах полученных в [73] (стр. 493 - 503).

Портфель $\pi = (\beta_k, \gamma_k)_{k=0}^N$ ценных бумаг называется **самофинансируемым**, если для него выполнено соотношение (1.1), а в формуле (а) теоремы 1.1 $h_k = 0$ при всех $k = 1, 2, \dots, N$. Это условие означает, что перед изменением состава портфеля в промежутке между моментами времени $k - 1$ и k портфель не испытывает ни притока дополнительного капитала, ни его оттока.

Все рассматриваемые в данной работе портфели, мы будем считать самофинансируемыми.

Определение 1.4 *Финансовым обязательством называется произвольная неотрицательная с.в. $F_N \geq 0$.*

Финансовое обязательство измеримо относительно σ -алгебры \mathcal{F}_N . Экономически финансовое обязательство — это количество денег, которое мы запланировали получить в терминальный момент времени N в результате ведения портфеля ценных бумаг. Процедура построения такого портфеля называется *хеджированием* данного обязательства F_N .

Определение 1.5 *(B, S) -рынок называется **полным**, если любое финансовое обязательство F_N реплицируемо, то есть существует такой самофинансируемый портфель $\pi = (\beta_k, \gamma_k)_{k=0}^N$, что его финальный капитал в точности совпадает с F_N : $X_N^\pi = F_N$.*

Определение 1.6 *(B, S) -рынок называется **безарбитражным**, если не существует самофинансируемого портфеля, удовлетворяющего условиям: $X_0 = 0$ и $X_N \geq 0$, причём хотя бы одно из чисел $X_N(B_i) > 0$, $B_i \in \Omega$, $i \in \mathbb{N}$.*

"Отсутствие арбитража" на рынке означает, что он является "честным" в том смысле, что на нём нет возможности получения прибыли без риска.

В диссертации мы будем рассматривать статические модели (т.е. $N = 1$). Пусть $\mathcal{F}_0 = \{\Omega, \emptyset\}$, а $\mathcal{F}_1 = \sigma(B_1, B_2, \dots)$ — σ -алгебра, порождённая разбиением Ω на счётное число атомов B_1, B_2, \dots рис. 3. Будем считать, что новые события возникают на рынке неодновремен-

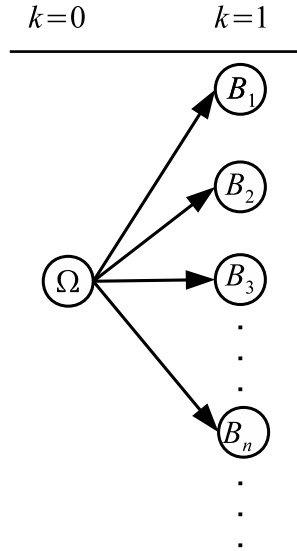


Рис. 3

но, а именно: в момент $k = 0$ мы имеем только один атом Ω , а при переходе от момента $k = 0$ к моменту $k = 1$ этот атом раздробился на счётное число атомов B_1, B_2, \dots . Не нарушая общности, можно считать, что порядок их появления таков: B_1, B_2, \dots . Вводить промежуточные времена можно по-разному, лишь бы они были упорядочены. Для этого определим следующие σ -алгебры событий: $\mathcal{H}_0 = \mathcal{F}_0$, $\mathcal{H}_1 = \{\Omega, \emptyset, B_1, \overline{B_1}\}$ (эта σ -алгебра порождена атомом B_1 , т.е. $\mathcal{H}_1 = \sigma\{B_1\}$). Далее полагаем $\mathcal{H}_2 = \sigma\{B_1, B_2\}, \dots, \mathcal{H}_\infty = \sigma\{B_1, B_2, \dots\}$. Ясно, что $\mathcal{H}_\infty = \mathcal{F}_1$. Полученная фильтрация $\{\Omega, \emptyset\} = \mathcal{H}_0 \subset \mathcal{H}_1 \subset \dots \subset \mathcal{H}_\infty = \mathcal{F}_1$ обладает следующим свойством: при переходе от n к $n + 1$ только один атом из \mathcal{H}_n дробится на две части, причём тот, который в результате дробления был получен на предыдущем шаге, а остальные атомы этой σ -алгебры остаются неизменными. Такие фильтрации называются спе-

циальными хааровскими фильтрациями. Так как начальная и конечная σ -алгебры построенной хааровской фильтрации совпадают с исходными σ -алгебрами \mathcal{F}_0 и \mathcal{F}_1 , то естественно говорить, что фильтрация $(\mathcal{H}_n)_{n=0}^\infty$ интерполирует фильтрацию $(\mathcal{F}_k)_{k=0}^1$. При этом исходная временная шкала как бы "растягивается" рис. 4. Прежде чем определить хааровскую

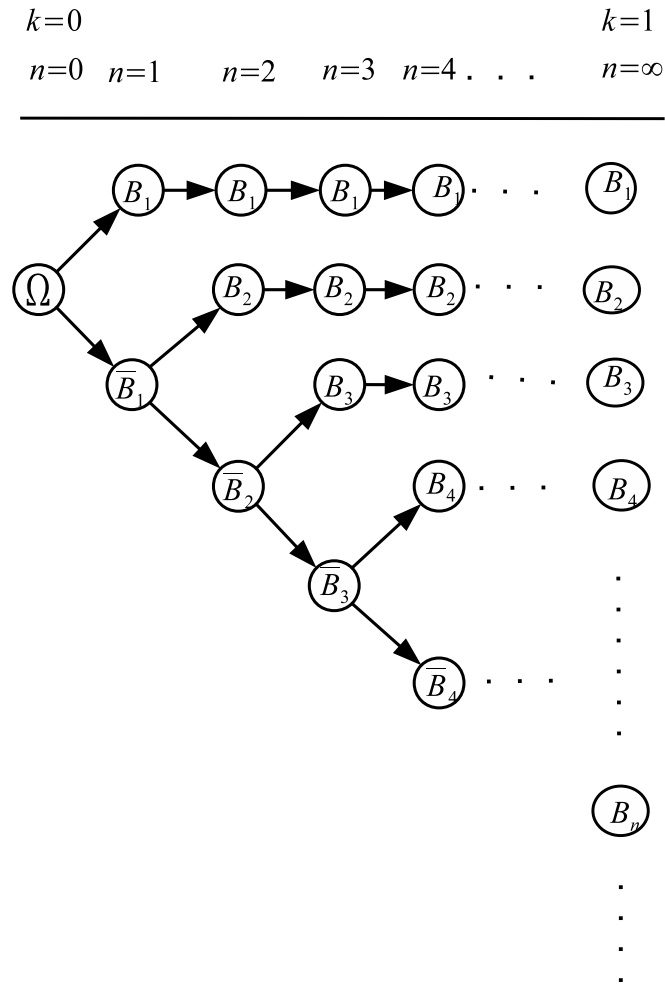


Рис. 4

интерполяцию цен акций, нам необходимо ввести понятие мартингала и мартингальной меры.

Пусть $\Omega = \bigcup_{i=1}^\infty B_i$. Обозначим значения случайного процесса Z следующим образом:

$$a := Z_0|_\Omega, \quad b_i := Z_1|_{B_i}, \quad a, b_i \in \mathbb{R}, \quad i \in \mathbb{N}. \tag{1.2}$$

Определение 1.7 *Вероятностная мера $P \in \mathcal{P}$ называется мартингальной мерой, если выполняется равенство:*

$$a = \sum_{i=1}^{\infty} b_i p_i .$$

При этом сам процесс $Z = (Z_k, \mathcal{F}_k, P)_{k=0}^1$ называется мартингалом.

Обозначим через $\mathcal{P}(Z, \mathbf{F})$ множество мартингальных мер с.п. Z . Нетрудно убедиться в том, что множество $\mathcal{P}(Z, \mathbf{F})$ является выпуклым подмножеством множества \mathcal{P} . Множество невырожденных мартингальных мер $\mathcal{P}(Z, \mathbf{F})$ совпадает с множеством решений системы:

$$\begin{cases} \sum_{i=1}^{\infty} p_i = 1 \\ \sum_{i=1}^{\infty} b_i p_i = a \\ p_i > 0, i \in \mathbb{N}. \end{cases} \quad (1.3)$$

Лемма 1.1 *Для того, чтобы $\mathcal{P}(Z, \mathbf{F}) \neq \emptyset$, необходимо и достаточно, чтобы выполнялись неравенства*

$$\inf_{1 \leq i < \infty} b_i < a < \sup_{1 \leq i < \infty} b_i , \quad (1.4)$$

либо выполнялись равенства

$$a = b_1 = b_2 = b_3 = \dots \quad (1.5)$$

В данной работе мы не рассматриваем тривиальный случай (1.5) и всюду предполагаем, что выполнено условие (1.4). Очевидно, что в этом случае система (1.3) имеет бесконечное число решений. Согласно 1-ой и 2-ой фундаментальным теоремам финансовой математики соответствующий рынок является безарбитражными и неполными.

Зафиксируем меру $P \in \mathcal{P}(Z, \mathbf{F})$. По мартингалу $Z = (Z_k, \mathcal{F}_k, P)_{k=0}^1$ построим мартингальную хааровскую интерполяцию $Y = (Y_n, \mathcal{H}_n, P)_{n=0}^{\infty}$:

$Y_n = E^P[Z_1 | \mathcal{H}_n]$ (мартингал Дирихле). Если специальная хааровская интерполирующая фильтрация \mathbf{H} фильтрации \mathbf{F} фиксирована, то мартингал Y определяется по мартингалу Z однозначно.

Лемма 1.2 1) Пусть $Y = (Y_n, \mathcal{H}_n)_{n=0}^\infty$ — \mathbf{H} -адаптированный случайный процесс. На (Ω, \mathcal{F}) существует мартингальная мера $P \in \mathcal{P}(Y, \mathbf{H})$ тогда и только тогда, когда $\forall n = 0, 1, 2, \dots$ выполняется равенство

$$Y_n|_{\Omega \setminus C} = Y_{n+1}|_{\Omega \setminus C} \quad (1.6)$$

и одно из следующих соотношений:

$$Y_{n+1}|_B < Y_n|_C < Y_{n+1}|_D, \quad (1.7)$$

$$Y_{n+1}|_B > Y_n|_C > Y_{n+1}|_D, \quad (1.8)$$

$$Y_{n+1}|_B = Y_n|_C = Y_{n+1}|_D, \quad (1.9)$$

где $C \in \mathcal{H}_n$, $C = B + D$, а B и D — атомы из \mathcal{H}_{n+1} .

2) Мартингальная мера $P \in \mathcal{P}(Y, \mathbf{H})$ существует и единственна тогда и только тогда, когда $\forall n = 0, 1, 2, \dots$ выполняется равенство (1.6) и одно из соотношений (1.7) или (1.8).

3) $|\mathcal{P}(Y, \mathbf{H})| = \infty$ тогда и только тогда, когда выполняются условия пункта 1) и $\exists n$, для которого выполняется равенство (1.9).

Определение 1.8 Будем говорить, что мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетворяет ослабленному свойству универсальной хааровской единственности (ОСУХЕ), если для любой специальной хааровской интерполяции \mathbf{H} исходной фильтрации \mathbf{F} и соответствующей хааровской интерполяции $Y = (Y_n, \mathcal{H}_n)_{n=0}^\infty$ мартингала $Z = (Z_k, \mathcal{F}_k, P)_{k=0}^N$ имеет место равенство $|\mathcal{P}(Y, \mathbf{H})| = 1$.

Определение 1.9 Будем говорить, что мартингальная мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетворяет ослабленному условию несовпадения барицентров (ОУНБ), если:

- 1) ряд $\sum_{i=1}^{\infty} b_i p_i$ сходится абсолютно;
- 2) $\forall i = 1, 2, \dots$ и для любого набора индексов $J \subset \{1, 2, \dots\}$ тако- го, что $i \notin J$; \bar{J} — конечно, выполняется неравенство:

$$b_i \neq \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j}. \quad (1.10)$$

Множество мартингальных мер, удовлетворяющих ОУНБ, будем обозна- чать через: $\text{ОУНБ}(Z)$.

Замечание 1.1 Если мартингальная мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетво- ряет ОУНБ, то справедливо неравенство:

$$b_i \neq a, \forall i \in \mathbb{N}.$$

Действительно, если в неравенстве (1.10) взять $J = \{1, 2, \dots\} \setminus \{i\}$, то

$$b_i \neq \frac{a - b_i p_i}{1 - p_i} \iff b_i \neq a.$$

Теорема 1.2 Мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетворяет ОСУХЕ тогда и только тогда, когда она удовлетворяет ОУНБ.

Необходимость. Предположим противное. Будем считать, что мартин- гальная мера P не удовлетворяет ОУНБ. Пусть существует индекс i и набор индексов $J = \{j_1, j_2, \dots\}$ ($i \notin J$) из определения 1.9 такие, что $b_i = \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j}$. Сконструируем с.х.и.ф. \mathbf{H} фильтрации $\mathbf{F} = \{\mathcal{F}_k\}_{k=0}^1$ следу- ющим образом: пусть $B = \bigcup_{j \in J} B_j$, $C := B_i + B$, $C \subset \Omega$ (случай, когда $C = \Omega$ рассматривается аналогично). Рассмотрим конечное множество индексов: $\overline{i \cup J} = \{i_1, i_2, \dots, i_\alpha\}$. Полагаем:

$$\mathcal{H}_{n_1} = \sigma\{B_{i_1}\}$$

$$\mathcal{H}_{n_2} = \sigma\{B_{i_1}, B_{i_2}\}$$

.....

.....

$$\mathcal{H}_{n_\alpha} = \sigma\{B_{i_1}, B_{i_2}, \dots, B_{i_\alpha}\}$$

$$\mathcal{H}_{n_\alpha+1} = \sigma\{B_{i_1}, B_{i_2}, \dots, B_{i_\alpha}, B_i\}$$

Продолжение построения с.х.и.ф. \mathbf{H} осуществляется произвольным образом. Очевидно, что $C \in \mathcal{H}_{n_\alpha}$. При переходе от момента времени n_α к моменту $n_\alpha + 1$ атом C дробится на атомы B_i и B . Применяя мартингалное свойство, получим:

$$\int_B Y_{n_\alpha} dP = \int_B Z_1 dP \Rightarrow Y_{n_\alpha}|_B P(B) = \sum_{j \in J} b_j p_j \Rightarrow Y_{n_\alpha}|_B = \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j}$$

Ясно также, что $Y_{n_\alpha}|_{B_i} = Z_1|_{B_i} = b_i$. Следовательно, $Y_{n_\alpha+1}|_{B_i} = Y_{n_\alpha+1}|_B$. Из пункта 3) леммы 1.1 вытекает равенство $|\mathcal{P}(Y, \mathbf{H})| = \infty$, то есть мера P не удовлетворяет ОСУХЕ.

Достаточность. Пусть выполняется ОУНБ и пусть $(\mathcal{H}_n)_{n=0}^\infty$ — произвольная с.х.и.ф. исходной фильтрации. Пусть $C \in \mathcal{H}_n$ — атом, дробящийся на атомы B_i и $B = \bigcup_{j \in J} B_j$. Обозначим $J = \{j_1, j_2, \dots\}$. Легко видеть, что \bar{J} — конечно (число элементов этого множества не превышает $n + 1$). Если выполнено ОУНБ (то есть выполнено неравенство (1.10)), то применяя рассуждения из доказательства необходимости, получаем $Y_n|_{B_i} \neq Y_n|_B$. По пункту 2) леммы 1.1 процесс $(Y_n, \mathcal{H}_n)_{n=0}^\infty$ обладает единственной мартингалльной мерой P . ■

1.2 Исследование (B, S)-рынков с помощью аргументов двойственности

Рассмотрим два лица, действующих на финансовом рынке. Одно из них — покупатель, он приобретает какой-нибудь контракт, платя за это величину C — цену (стоимость) контракта. Получив деньги C , другое лицо — продавец — должен гарантировать покупателю выполнение этого контракта в какой-то момент времени N . Для этого он создаёт портфель $\pi = (\beta_n, \gamma_n)_{n=0}^N$ с начальным капиталом $X_0^\pi = C$ и задаётся целью получить максимальный финальный капитал X_N^π . Сумма X_N^π должна быть по меньшей мере такова, чтобы обязательства по контракту могли быть выполнены. Контракты бывают **европейского типа** и **американского типа**. Первые характеризуются тем, что исполняются точно в финальный момент времени N . Контракты же американского типа могут быть предъявлены к исполнению (погашены) в любой момент времени до момента N включительно. Мы будем работать только с контрактами европейского типа. Одним из основных вопросов, которые решает стохастическая финансовая математика, является вопрос о том: какова стоимость C контракта, которая, с одной стороны, давала бы возможность продавцу контракта построить стратегию, приводящую к достижению финансового обязательства и выполнению контракта, а с другой — устраивала бы покупателя контракта. Речь идёт о **справедливой цене опциона**. Для решения данной задачи, нам понадобятся следующие определения и факты.

Определение 1.10 Самофинансируемый портфель $\pi = (\beta_k, \gamma_k)_{k=0}^N$ называется **верхним (нижним) (x, F_N) -хеджем**, если он удовлетворяет условиям:

$$1) X_0^\pi = x;$$

2) $X_N^\pi \geq F_N$ ($X_N^\pi \leq F_N$). Совокупность всех верхних (нижних) (x, F_N) -хеджесей обозначается $\Pi^*(x, F_N)$ ($\Pi_*(x, F_N)$).

Продавец контракта, получивший в начальный момент времени премию C , строит верхний (x, F_N) -хедж для того, чтобы выполнить финансовое обязательство.

Определение 1.11 *Верхней ценой (нижней ценой) контракта называется число*

$$C^*(F_N) = \inf\{x : \Pi^*(x, F_N) \neq \emptyset\},$$

соответственно

$$C_*(F_N) = \sup\{x : \Pi_*(x, F_N) \neq \emptyset\}.$$

Понятно, что для любого финансового обязательства F_N существует такой начальный капитал x , что $\Pi^*(x, F_N) \neq \emptyset$. Например, достаточно взять $x = \sup_{1 \leq i < \infty} F_N(B_i)$.

Определение 1.12 (x, F_N) -хедж называется совершенным, если он является одновременно верхним и нижним, то есть выполняется равенство $X_N^\pi = F_N$.

Теорема 1.3 *Если данный (\mathbf{B}, \mathbf{S}) -рынок безарбитражен, то для любого финансового обязательства F_N верно неравенство $C_*(F_N) \leq C^*(F_N)$.*

Теорема 1.4 *Если данный (\mathbf{B}, \mathbf{S}) -рынок безарбитражен и полон, то для любого финансового обязательства F_N выполняется $C_*(F_N) = C^*(F_N)$, причём существует совершенный (x, F_N) -хедж такой, что $x = C_*(F_N) = C^*(F_N)$.*

Вернёмся к рассмотрению одношагового ($N = 1$) (\mathbf{B}, \mathbf{S}) -рынка, который является безарбитражным и неполным. Пусть $F_1 = \sum_{i=1}^{\infty} c_i I_{B_i}$ — финансовое обязательство. Обозначим через: $\bar{\mathcal{P}} = \{P = (p_1, p_2, \dots) : p_i \geq 0, \forall i = 1, 2, \dots\}$, $\bar{\mathcal{P}}(Z, \mathbf{F}) = \{P \in \bar{\mathcal{P}} : \sum_{i=1}^{\infty} p_i = 1, \sum_{i=1}^{\infty} b_i p_i = a\}$ ($\mathcal{P}(Z, \mathbf{F}) \subset \bar{\mathcal{P}}(Z, \mathbf{F})$). Пусть $P \in \bar{\mathcal{P}}(Z, \mathbf{F})$. Цена финансового обязательства определяется по формуле:

$$C(F_1) = E^P[F_1] = \sum_{i=1}^{\infty} c_i p_i.$$

Вычисление нижней и верхней цен финансового обязательства, осуществляется решением задач линейного программирования с учётом формул:

$$C_*(F_1) = \inf_{P \in \bar{\mathcal{P}}(Z, \mathbf{F})} E^P[F_1], \quad (1.11)$$

$$C^*(F_1) = \sup_{P \in \bar{\mathcal{P}}(Z, \mathbf{F})} E^P[F_1]. \quad (1.12)$$

Построим двойственные задачи к задачам (1.11), (1.12). Рассмотрим систему неравенств:

$$\begin{cases} u_1 + b_1 u_2 \geq c_1 \\ u_1 + b_2 u_2 \geq c_2 \\ \dots \\ u_1 + b_j u_2 \geq c_j \\ \dots \\ u_1, u_2 \in \mathbb{R}. \end{cases} \quad (1.13)$$

Множество решений этой системы обозначим через: $\bar{U} = \{U = (u_1, u_2) : u_1 + b_j u_2 \geq c_j, j = 1, 2, \dots\}$. Для $\forall U \in \bar{U}$ рассмотрим значение $g(U) = u_1 + a u_2$. Тогда двойственной задачей к задаче (1.11) является:

$$L^* = \sup_{U \in \bar{U}} \{g(U)\}, \quad (1.14)$$

а двойственной к задаче (1.12) является:

$$L_* = \inf_{U \in \bar{U}} \{g(U)\}. \quad (1.15)$$

Определение 1.13 *Говорят, что задача (1.12) (соответственно (1.15) разрешима, если $\exists P \in \bar{\mathcal{P}}(Z, \mathbf{F}) : E^P[F_1] = C^*(F_1) < +\infty$ ($\exists U \in \bar{U} : g(U) = L_* > -\infty$).*

Будем считать, что $C^*(F_1) = -\infty$, если $\bar{\mathcal{P}}(Z, \mathbf{F}) = \emptyset$ и $L_* = +\infty$, если $\bar{U} = \emptyset$.

Замечание 1.2 $C^*(F_1) \leq L_*$.

Действительно $C^* = \sup_{P \in \bar{\mathcal{P}}(Z, \mathbf{F})} \sum_{i=1}^{\infty} c_i p_i \leq \inf_{U \in \bar{U}} \sum_{i=1}^{\infty} (u_1 + b_i u_2) p_i = \inf_{U \in \bar{U}} u_1 \sum_{i=1}^{\infty} p_i + u_2 \sum_{i=1}^{\infty} b_i p_i = \inf_{U \in \bar{U}} \{u_1 + u_2 a\} = L_*$.

Определение 1.14 *Неравенство*

$$g(U) = u_1 + a u_2 \geq L \quad (1.16)$$

является следствием совместной системы линейных неравенств (1.13), если произвольное решение этой системы является решением неравенства (1.16).

Определение 1.15 *Система неравенств (1.13) называется финитно определённой, если из того, что линейное неравенство (1.16) является следствием системы (1.13) следует, что оно является следствием и некоторой её конечной подсистемы.*

В дальнейшем мы будем предполагать, что задача (1.15) имеет финитно определённую систему ограничений. Рассмотрим случай, когда конечная подсистема в определении 1.15 состоит из первых $s \geq 2$ неравенств системы (1.13). Множество решений этой системы обозначим через:

$$\bar{U}_s := \{U = (u_1, u_2) : u_1 + b_j u_2 \geq c_j, j = 1, 2, \dots, s\}.$$

Пусть $t \in \mathbb{R}$, $t \geq 0$, обозначим через:

$$\bar{U}_s(t) := \{U = (u_1, u_2) : u_1 + b_j u_2 \geq c_j t, j = 1, 2, \dots, s\}.$$

Рассмотрим систему векторов: $\{(1, a), (1, b_1), \dots, (1, b_s)\}$. Максимальное число линейно независимых векторов среди них равно двум. Предположим, что $b_1 < a < b_2$, $\{(1, b_1), (1, b_2)\}$ — линейно независимая система векторов. Пусть $\bar{U}_2 := \{U = (u_1, u_2) : u_1 + b_j u_2 \geq c_j, j = 1, 2\}$, $\bar{U}_2(t) := \{U = (u_1, u_2) : u_1 + b_j u_2 \geq c_j t, j = 1, 2, t \geq 0\}$.

Лемма 1.3 *Неравенство $g(U) \geq L$ является следствием некоторой совместной конечной подсистемы $\bar{U}_2 \Leftrightarrow$ неравенство $g(U) = u_1 + a u_2 \geq L_* t$ является следствием подсистемы $\bar{U}_2(t)$.*

Необходимость. Пусть (u_1^0, u_2^0, t) — решение системы $\bar{U}_2(t)$. Пусть $t > 0$, тогда $u_1^0 + b_j u_2^0 \geq c_j t$, $j = 1, 2$. Значит $\frac{u_1^0}{t_0} + b_j \frac{u_2^0}{t_0} \geq c_j$, $j = 1, 2$ и $(\frac{u_1^0}{t_0}, \frac{u_2^0}{t_0})$ — решение системы \bar{U}_2 . Так как $g(U) \geq L$ является следствием системы \bar{U}_2 , поэтому $(\frac{u_1^0}{t_0}, \frac{u_2^0}{t_0})$ удовлетворяет неравенству $g(U) \geq L_*$, то есть $\frac{u_1^0}{t_0} + a \frac{u_2^0}{t_0} \geq L_*$ или $g(u_1^0, u_2^0) \geq L_* t_0$. Отсюда вытекает, что (u_1^0, u_2^0, t_0) удовлетворяет неравенству $g(U) \geq L_* t$, то есть $g(U) \geq L_* t$ является следствием $\bar{U}_2(t)$.

Пусть (u_1^0, u_2^0, t) — решение системы $\bar{U}_2(t)$. Пусть теперь $t = 0$. Возьмём ещё некоторое решение $(\bar{u}_1, \bar{u}_2, \bar{t})$, $t > 0$ системы $\bar{U}_2(t)$ (например $(\bar{u}_1, \bar{u}_2, 1)$, где (u_1, u_2) — произвольное решение системы \bar{U}_2). Обозначим $u_1(\gamma) = (1 - \gamma)\bar{u}_1 + \gamma u_1^0$, $u_2(\gamma) = (1 - \gamma)\bar{u}_2 + \gamma u_2^0$, $t(\gamma) = (1 - \gamma)\bar{t} + \gamma t_0$, $\forall \gamma \in (0, 1)$. Ясно, что вектор $(u_1(\gamma), u_2(\gamma), t(\gamma))$, $\forall \gamma \in (0, 1)$ является решением системы $\bar{U}_2(t)$. Но так как $t(\gamma) > 0$, то по рассмотренному выше случаю $g((u_1(\gamma), u_2(\gamma))) \geq L_* t(\gamma)$. Поскольку $g(u_1(\gamma), u_2(\gamma)) = (1 - \gamma)\bar{u}_1 + \gamma u_1^0 + a((1 - \gamma)\bar{u}_2 + \gamma u_2^0) = (1 - \gamma)(\bar{u}_1 + a\bar{u}_2) + \gamma(u_1^0 + a u_2^0) \geq$

$L_*((1-\gamma)\bar{t} + \gamma t_0) = (1-\gamma)L_*\bar{t} + \gamma L_*t_0$, то переходя в последнем равенстве к пределу при $\gamma \rightarrow 1 - 0$, получим, что $g(u_1^0, u_2^0) \geq K_*t_0$. Таким образом, решение (u_1^0, u_2^0, t_0) системы $\bar{U}_2(t)$ при $t_0 = 0$ удовлетворяет неравенству $g(U) \geq L_*t$.

Достаточность условий очевидна. ■

Лемма 1.4 *Неравенство $g(U) \geq L_*$ является следствием некоторой совместной конечной подсистемы $\bar{U}_2 \Leftrightarrow \exists p_i \geq 0, i = 1, 2$:*

$$\begin{cases} p_1 + p_2 = 1 \\ b_1p_1 + b_2p_2 = a \\ c_1p_1 + c_2p_2 = L. \end{cases}$$

Необходимость. Пусть $g(U) \geq L_*$ является следствием некоторой совместной конечной подсистемы ограничений \bar{U}_2 . Рассмотрим ситему

$$\begin{cases} p_1 + p_2 = 1 \\ b_1p_1 + b_2p_2 = a \\ p_i \geq 0, i = 1, 2. \end{cases} \quad (1.17)$$

Так как $\{(1, b_1), (1, b_2)\}$ — линейно независимая система векторов и $b_1 < a < b_2$, поэтому система (1.17) имеет единственное решение. Пусть (u_1, u_2) — некоторое решение системы

$$\begin{cases} u_1 + b_1p_1 \geq c_1 \\ u_1 + b_2p_1 \geq c_2. \end{cases}$$

Тогда для него выполняется неравенство $u_1 + au_2 \geq L_*$. Если в последней системе первое неравенство умножить на p_1 , а второе на p_2 и сложить, тогда получим: $p_1u_1 + p_2u_1 + b_1u_2p_1 + b_2u_2p_2 = u_1(p_1 + p_2) + u_2(b_1p_1 + b_2p_2) = u_1 + au_2 \geq c_1p_1 + c_2p_2 = L_*$. Поэтому решение системы (1.17) является

решением системы:

$$\begin{cases} p_1 + p_2 = 1 \\ b_1 p_1 + b_2 p_2 = a \\ c_1 p_1 + c_2 p_2 = L \\ p_i \geq 0, \quad i = 1, 2. \end{cases}$$

Достаточность очевидна. ■

Следствие 1.1 *Неравенство $g(U) \geq L_*$ является следствием некоторой совместной конечной подсистемы $\bar{U}_2 \Leftrightarrow \exists p_i \geq 0, \quad i = 1, 2$:*

$$g(U) - L_* = \sum_{i=1}^2 p_i (u_1 + b_i u_2 - c_i) .$$

Следствие 1.2 *Неравенство $g(U) \geq L_*$ является следствием некоторой совместной конечной подсистемы $\bar{U}_2 \Leftrightarrow \exists p_i \geq 0, \quad i = 1, 2$:*

$$g(U) - L_* t = \sum_{i=1}^2 p_i (u_1 + b_i u_2 - c_i) + p_0 t .$$

Теорема 1.5 *Пусть система (1.13) является финитно определённой. Если $L_* > -\infty$, то задача (1.12) разрешима, при этом $C^* = L_*$.*

Доказательство. Пусть \bar{U} — финитно определённая система. Так как неравенство $g(U) \geq L_*$ является следствием системы (1.13), то $g(U) \geq L_*$ является следствием и некоторой конечной подсистемы ограничений \bar{U}_2 . Тогда в силу леммы 1.4 получим, что $\exists p_i \geq 0, \quad i = 1, 2$:

$$\begin{cases} p_1 + p_2 = 1 \\ b_1 p_1 + b_2 p_2 = a \\ c_1 p_1 + c_2 p_2 = L_* . \end{cases}$$

Пусть $P = (p_1, p_2, \dots, p_i, \dots)$, причём $p_i = 0$, $i \neq 1, 2$, тогда получим:

$$\begin{cases} p_1 + p_2 + \dots = 1 \\ b_1 p_1 + b_2 p_2 + \dots = a \\ c_1 p_1 + c_2 p_2 + \dots = L_*. \end{cases}$$

Значит $P \in \overline{\mathcal{P}}(Z, \mathbf{F})$ и $L_* = c_1 p_1 + c_2 p_2 + \dots \leq C^*(F_1)$. Согласно замечанию 1.2: $C^*(F_1) \leq L_*$. Поэтому $C^*(F_1) = L_*$. ■

Следствие 1.3 Пусть система (1.13) является финитно определённой. Если задача (1.12) разрешима, то $L_* > -\infty$. При этом справедливо равенство: $C^*(F_1) = L_*$.

Пусть задача (1.12) разрешима, тогда $\exists P^* \in \overline{\mathcal{P}}(Z, \mathbf{F})$: $E^{P^*}[F_1] = C^*(F_1) < \infty$. Поэтому $\overline{\mathcal{P}}(Z, \mathbf{F}) \neq \emptyset$ и $-\infty < C^*(F_1) < +\infty$. Используя замечание 1.2 и теорему 1.5, получим $C^*(F_1) = L_*$.

Пример 1.1 Пусть $a = 3$, $b_i = i(i + 1)$, $c_i = 2i + 1$, $i = 1, 2, \dots$.

В этом случае задача (1.12) выглядит следующим образом:

$$E^P[F_1] = 3p_1 + 5p_2 + 7p_3 + \dots \rightarrow \sup$$

$$\begin{cases} p_1 + p_2 + \dots = 1 \\ 2p_1 + 6p_2 + \dots = 3 \\ p_i \geq 0, \quad i = 1, 2, \dots \end{cases}$$

Соответственно двойственная к ней задача :

$$g(U) = u_1 + 3u_2 \rightarrow \inf$$

$$\begin{cases} u_1 + 2u_2 \geq 3 \\ u_1 + 6u_2 \geq 5 \\ u_1 + 12u_2 \geq 7 \\ \dots \end{cases}$$

Цена финансового обязательства равна $C^*(F_1) = L_* = 3,5$ при $u_1 = 2$, $u_2 = 0,5$. Соответствующая мартингальная мера $P = (\frac{3}{4}, \frac{1}{4}, 0, 0, \dots)$.

1.3 Общий алгоритм проведения финансовых расчётов на одношаговых (B, S) -рынках со счётным числом состояний

В этом параграфе представлена последовательность шагов, позволяющих реплицировать финансовое обязательство, заданное на безарбитражном финансовом $(1, Z)$ -рынке. Это обобщённая модель алгоритма, который использован в программном комплексе, подробное описание которого будет представлено в главе 3.

Шаг 1. Проверим, реплицируемо ли заданное финансовое обязательство $F_1 = \sum_{i=1}^{\infty} c_i I_{B_i}$ на исходном финансовом рынке. Для этого рассмотрим систему:

$$\begin{cases} \beta_1 + \gamma_1 Z_1(B_1) = c_1 \\ \beta_1 + \gamma_1 Z_1(B_2) = c_2 \\ \dots \\ \beta_1 + \gamma_1 Z_0(\Omega) = X_0 \end{cases} \quad (1.18)$$

при этом $X_0 = \beta_0 + \gamma_0 Z_0(\Omega)$ ($X_0 = \beta_0$ при $\gamma_0 = 0$). Если система (1.18) имеет решение, тогда определен самофинансируемый портфель $\pi = (\beta_k, \gamma_k)_{k=0}^1$ такой, что его полный капитал X_1^π в точности равен F_1 . Переходим на шаг 5.

В случае, если рассматриваемая система (1.18) не имеет решение, то финансовое обязательство F_1 нереплицируемо. Переходим на шаг 2.

Шаг 2. С учётом значений с.п. Z , конструируем мартингальную меру $P \in \mathcal{P}(Z, \mathbf{F})$, удовлетворяющую ОСУХЕ (соответствующие формулы

вычисления таких мартингалльных мер представлены в главе 2). С помощью полученной меры, определяем справедливую цену финансового обязательства:

$$C = \sum_{i=1}^{\infty} c_i p_i .$$

Полученная цена удовлетворяет неравенству:

$$C_*(F_1) < C < C^*(F_1),$$

где $C_*(F_1), C^*(F_1)$ — решения задач (1.11), (1.12). Переходим на шаг 3.

Шаг 3. С помощью мартингалльной меры P , удовлетворяющей ОСУХЕ, строим интерполирующую фильтрацию исходного $(1, Z)$ -рынка. Рассмотрим некоторую перестановку натуральных чисел: $\delta = (\delta^{(1)}, \delta^{(2)}, \dots)$. Координаты вектора δ — зависимые случайные величины, определяемые с помощью генератора случайных чисел, основанном на геометрическом распределении [16]. Пусть \mathbf{H} — с.х.и.ф. фильтрации \mathbf{F} (рис. 2). Интерполяция процесса Z представляет собой:

$$\begin{aligned} Y_n &= E^P[Z_1 | \mathcal{H}_n] = \sum_{k=1}^n \frac{1}{P(B_{\delta^{(k)}})} E^P[Z_1 I_{B_{\delta^{(k)}}}] I_{B_{\delta^{(k)}}} + \\ &\frac{1}{P(\bar{B}_{\delta^{(n)}})} E^P[Z_1 I_{\bar{B}_{\delta^{(n)}}}] I_{\bar{B}_{\delta^{(n)}}} = \sum_{k=1}^n \frac{1}{P(B_{\delta^{(k)}})} E^P\left[\sum_{i=1}^{\infty} b_{\delta^{(i)}} I_{B_{\delta^{(i)}}} I_{B_{\delta^{(k)}}}\right] I_{B_{\delta^{(k)}}} + \\ &\frac{1}{P(\bar{B}_{\delta^{(n)}})} E^P\left[\sum_{i=1}^{\infty} b_{\delta^{(i)}} I_{B_{\delta^{(i)}}} I_{\bar{B}_{\delta^{(n)}}}\right] I_{\bar{B}_{\delta^{(n)}}} = \sum_{k=1}^n \frac{1}{P(B_{\delta^{(k)}})} \sum_{i=1}^{\infty} b_{\delta^{(i)}} E^P[I_{B_{\delta^{(i)}}} I_{B_{\delta^{(k)}}}] I_{B_{\delta^{(k)}}} + \\ &\frac{1}{P(\bar{B}_{\delta^{(n)}})} \sum_{i=1}^{\infty} b_{\delta^{(i)}} E^P[I_{B_{\delta^{(i)}}} I_{\bar{B}_{\delta^{(n)}}}] I_{\bar{B}_{\delta^{(n)}}} = \sum_{k=1}^n \frac{1}{P(B_{\delta^{(k)}})} \sum_{i=1}^{\infty} b_{\delta^{(i)}} P(B_{\delta^{(i)}} B_{\delta^{(k)}}) I_{B_{\delta^{(k)}}} + \\ &\frac{1}{P(\bar{B}_{\delta^{(n)}})} \sum_{i=1}^{\infty} b_{\delta^{(i)}} P(B_{\delta^{(i)}} \bar{B}_{\delta^{(n)}}) I_{\bar{B}_{\delta^{(n)}}} = \sum_{k=1}^n \frac{1}{P(B_{\delta^{(k)}})} b_{\delta^{(k)}} P(B_{\delta^{(k)}}) I_{B_{\delta^{(k)}}} + \\ &\frac{1}{\sum_{k=n+1}^{\infty} P(B_{\delta^{(k)}})} \sum_{k=n+1}^{\infty} b_{\delta^{(k)}} P(B_{\delta^{(k)}}) I_{\bar{B}_{\delta^{(n)}}} = \sum_{k=1}^n b_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \frac{\sum_{k=n+1}^{\infty} b_{\delta^{(k)}} p_{\delta^{(k)}}}{\sum_{k=n+1}^{\infty} p_{\delta^{(k)}}} I_{\bar{B}_{\delta^{(n)}}} . \end{aligned}$$

Если обозначить через: $\bar{b}_{\delta^{(n)}} = \frac{\sum_{k=n+1}^{\infty} b_{\delta^{(k)}} p_{\delta^{(k)}}}{\sum_{k=n+1}^{\infty} p_{\delta^{(k)}}$, тогда окончательно

получим:

$$Y_n = \sum_{k=1}^n b_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{b}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}}.$$

Так как выполнены равенства: $Z_0 = Y_0$, $Z_1 = \lim_{n \rightarrow \infty} Y_n$ P -п.н., поэтому процесс $Y = (Y_n, \mathcal{H}_n)_{n=0}^{\infty}$ интерполирует с.п. Z и обладает единственной мартингальной мерой, равной исходной мере P .

Шаг 4. Рассчитаем компоненты хеджирующего портфеля $\pi = (\beta_n, \gamma_n)_{n=0}^{\infty}$ на интерполирующем рынке.

$$X_n^\pi = E^P[F_1 | \mathcal{H}_n] = \sum_{k=1}^n c_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \frac{\sum_{k=n+1}^{\infty} c_{\delta^{(k)}} p_{\delta^{(k)}}}{\sum_{k=n+1}^{\infty} p_{\delta^{(k)}}} I_{\bar{B}_{\delta^{(n)}}}.$$

Обозначим через $\bar{c}_{\delta^{(n)}} = \frac{\sum_{k=n+1}^{\infty} c_{\delta^{(k)}} p_{\delta^{(k)}}}{\sum_{k=n+1}^{\infty} p_{\delta^{(k)}}}$. Тогда:

$$X_n^\pi = \sum_{k=1}^n c_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{c}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}}.$$

Запишем два условия:

$$X_n^\pi = \beta_n + \gamma_n Y_n; \quad X_{n-1}^\pi = \beta_n + \gamma_n Y_{n-1}.$$

Тогда с одной стороны:

$$\Delta X_n^\pi = X_n^\pi - X_{n-1}^\pi = \gamma_n (Y_n - Y_{n-1}) = \gamma_n \Delta Y_n, \quad (1.19)$$

а с другой:

$$\begin{aligned} \Delta X_n^\pi &= X_n^\pi - X_{n-1}^\pi = \sum_{k=1}^n c_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{c}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}} - \left(\sum_{k=1}^{n-1} c_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{c}_{\delta^{(n-1)}} I_{\bar{B}_{\delta^{(n-1)}}} \right) \\ &= c_{\delta^{(n)}} I_{B_{\delta^{(n)}}} + \bar{c}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}} - \bar{c}_{\delta^{(n-1)}} I_{\bar{B}_{\delta^{(n-1)}}} = c_{\delta^{(n)}} I_{B_{\delta^{(n)}}} + \bar{c}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}} - \bar{c}_{\delta^{(n-1)}} (I_{B_{\delta^{(n)}}} + I_{\bar{B}_{\delta^{(n-1)}}}) \\ &= (c_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}) I_{B_{\delta^{(n)}}} + (\bar{c}_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}) I_{\bar{B}_{\delta^{(n)}}}. \end{aligned} \quad (1.20)$$

$$\begin{aligned}\Delta Y_n = Y_n - Y_{n-1} &= \sum_{k=1}^n b_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{b}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}} - \left(\sum_{k=1}^{n-1} b_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{b}_{\delta^{(n-1)}} I_{\bar{B}_{\delta^{(n-1)}}} \right) = \\ &= (b_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}) I_{B_{\delta^{(n)}}} + (\bar{b}_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}) I_{\bar{B}_{\delta^{(n)}}}. \quad (1.21)\end{aligned}$$

Пусть

$$\gamma_n = \sum_{k=1}^{n-1} \gamma_n^{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \gamma_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}}. \quad (1.22)$$

Подставим (1.20), (1.21), (1.22) в (1.19):

$$\begin{aligned}(c_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}) I_{B_{\delta^{(n)}}} + (\bar{c}_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}) I_{\bar{B}_{\delta^{(n)}}} &= \left(\sum_{k=1}^{n-1} \gamma_n^{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \right. \\ &\left. \gamma_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}} \right) \left((b_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}) I_{B_{\delta^{(n)}}} + (\bar{b}_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}) I_{\bar{B}_{\delta^{(n)}}} \right) = \gamma_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}} (b_{\delta^{(n)}} - \\ &\bar{b}_{\delta^{(n-1)}}) I_{B_{\delta^{(n)}}} + \gamma_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}} (\bar{b}_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}) I_{\bar{B}_{\delta^{(n)}}} = \gamma_n^{\delta^{(n)}} (b_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}) I_{B_{\delta^{(n)}}} + \\ &\gamma_n^{\delta^{(n)}} (\bar{b}_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}) I_{\bar{B}_{\delta^{(n)}}}\end{aligned}$$

Последнее равенство возможно тогда и только тогда, когда:

$$\begin{cases} (c_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}) = \gamma_n^{\delta^{(n)}} (b_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}); \\ (\bar{c}_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}) = \gamma_n^{\delta^{(n)}} (\bar{b}_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}). \end{cases}$$

Из последней системы находим:

$$\gamma_n^{\delta^{(n)}} = \frac{c_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}}{b_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}} = \frac{\bar{c}_{\delta^{(n)}} - \bar{c}_{\delta^{(n-1)}}}{\bar{b}_{\delta^{(n)}} - \bar{b}_{\delta^{(n-1)}}}, \quad n = 1, 2, \dots$$

При этом будем считать, что $\gamma_n^{\delta^{(k)}} = 0$, $k = 1, 2, \dots, n-1$.

$$\begin{aligned}\gamma_n Y_n &= \left(\sum_{k=1}^{n-1} \gamma_n^{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \gamma_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}} \right) \left(\sum_{k=1}^n b_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{b}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}} \right) = \\ &\gamma_n^{\delta^{(n)}} b_{\delta^{(n)}} I_{B_{\delta^{(n)}}} + \gamma_n^{\delta^{(n)}} \bar{b}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}}.\end{aligned}$$

Согласно финальному условию, имеем:

$$\begin{aligned}\beta_n &= X_n^\pi - \gamma_n Y_n = \sum_{k=1}^n c_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \bar{c}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}} - \\ &(\gamma_n^{\delta^{(n)}} b_{\delta^{(n)}} I_{B_{\delta^{(n)}}} + \gamma_n^{\delta^{(n)}} \bar{b}_{\delta^{(n)}} I_{\bar{B}_{\delta^{(n)}}}) = \sum_{k=1}^{n-1} c_{\delta^{(k)}} I_{B_{\delta^{(k)}}} + (c_{\delta^{(n)}} - \gamma_n^{\delta^{(n)}} b_{\delta^{(n)}}) I_{B_{\delta^{(n)}}} + \\ &(\bar{c}_{\delta^{(n)}} - \gamma_n^{\delta^{(n)}} \bar{b}_{\delta^{(n)}}) I_{\bar{B}_{\delta^{(n)}}}.\end{aligned}$$

С учётом того, что:

$$\beta_n = \sum_{k=1}^{n-1} \beta_n^{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \beta_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}} = \sum_{k=1}^{n-1} \beta_n^{\delta^{(k)}} I_{B_{\delta^{(k)}}} + \beta_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}} + \beta_n^{\delta^{(n)}} I_{\bar{B}_{\delta^{(n-1)}}}.$$

Получим:

$$\begin{cases} \beta_n^{\delta^{(k)}} = c_{\delta^{(k)}}, k = 1, 2, \dots, n - 1 ; \\ \beta_n^{\delta^{(n)}} = c_{\delta^{(n)}} - \gamma_n^{\delta^{(n)}} b_{\delta^{(n)}} ; \\ \beta_n^{\delta^{(n)}} = \bar{c}_{\delta^{(n)}} - \gamma_n^{\delta^{(n)}} \bar{b}_{\delta^{(n)}}. \end{cases}$$

При этом выполняется процесс стабилизации:

$$\beta_n = \beta_n^{\delta^{(n)}} = \beta_n^{\delta^{(n+1)}} = \dots \quad \forall n = 1, 2, \dots,$$

$$\gamma_n = \gamma_n^{\delta^{(n)}} = \gamma_n^{\delta^{(n+1)}} = \dots, \quad \forall n = 1, 2, \dots$$

Легко видеть, что с.п. $X = (X_n^\pi, \mathcal{H}_n)_{n=0}^\infty$ является полным капиталом самофинансируемого портфеля $\pi = (\beta_n, \gamma_n)_{n=0}^\infty$. Выполнение равенства :

$$\lim_{n \rightarrow \infty} X_n^\pi = F_1 \text{ } P\text{-п.н.}$$

означает, что ф.о. F_1 реплицируемо, а интерполирующий рынок является полным.

Шаг 5. Конец алгоритма.

В заключение отметим о том, что рассматривая неполные модели (\mathbf{B}, \mathbf{S}) -рынков со счётным числом состояний, одной из основных задач является отыскание мартингальных мер, удовлетворяющих ОСУХЕ. Именно этому вопросу посвящена вторая глава диссертации.

Глава 2

Условия существования мартингальных мер, удовлетворяющих ослабленному свойству универсальной хааровской единственности, в случае счётного вероятностного пространства

Вторая глава состоит из четырёх параграфов.

В *параграфе 2.1* изложена общая классификация значений с.п. Z , в зависимости от которых получены различные условия существования мартингальных мер, удовлетворяющих ослабленному свойству универсальной хааровской единственности ОСУХЕ.

В *параграфе 2.2* указаны условия существования мартингальных мер, удовлетворяющих ОСУХЕ, для случая, когда с.в. Z_1 принимает конечное (не меньшее трёх) число различных значений.

В *параграфе 2.3* представлен конструктивистский подход получения мартингальных мер, удовлетворяющих ОУНБ, когда с.в. Z_1 принимает три различных значения.

В *параграфе 2.4* представлено достаточное условие существования мартингалльных мер, удовлетворяющих ОСУХЕ, когда с.в. Z_1 принимает бесконечное число различных значений.

2.1 Выполнение ОУНБ для различных моделей одношаговых рынков с бесконечным числом состояний

Рассмотрим фильтрованное пространство (Ω, \mathbf{F}) , где Ω — счётное множество, а $\mathbf{F} = (\mathcal{F}_k)_{k=0}^1$ — одношаговая фильтрация, в которой $\mathcal{F}_0 = \{\Omega, \emptyset\}$, а \mathcal{F}_1 — σ -алгебра, порождённая разбиением Ω на счётное число атомов $B_i, i \in \mathbb{N}$. Пусть $Z = (Z_k, \mathcal{F}_k)_{k=0}^1$ — \mathbf{F} адаптированный с.п. на (Ω, \mathbf{F}) .

Рассмотрим систему (1.3). Будем предполагать, что выполнено неравенство (1.4), т.е. рассматриваемый рынок является безарбитражным и неполным. Обозначим через $\text{ОУНБ}(Z)$ — множество мартингалльных мер, удовлетворяющих ОУНБ.

Замечание 2.1 Пусть $c \neq 0$ и d — произвольные действительные числа. Тогда $\mathcal{P}(cZ + d, \mathbf{F}) = \mathcal{P}(Z, \mathbf{F})$ и $\text{ОУНБ}(cZ + d) = \text{ОУНБ}(Z)$.

Определение 2.1 Будем говорить, что число b , входящее в последовательность $\{b_i\}_{i=1}^\infty$, имеет кратность m (m может быть как конечно, так и бесконечно), если в этой последовательности оно присутствует m раз.

Предложение 2.1 Если среди элементов последовательности $\{b_i\}_{i=1}^\infty$ конечное число значений имеет конечную кратность и одно значение имеет бесконечную кратность, тогда ни одна мера $P \in \mathcal{P}(Z, \mathbf{F})$ не удовлетворяет ОУНБ.

Доказательство. Не нарушая общности, будем считать, что $\exists l \in \mathbb{N}$: $b_l \neq b_{l-i}, i = 1, 2, \dots, l-1$ и $b_l = b_{l+1} = b_{l+2} = \dots$. Возьмём $i = l$, а $J = \{l+1, l+2, \dots\}$ (\bar{J} -конечно), тогда:

$$b_l \neq \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j} = b_l.$$

ОУНБ не выполнено. ■

Таким образом, если рассматриваемый (\mathbf{B}, \mathbf{S}) -рынок удовлетворяет условиям предложения 2.1, то для него нельзя построить интерполирующий его полный рынок.

В следующих результатах представлены условия, при выполнении которых всегда можно построить полный рынок, интерполирующий исходный неполный рынок.

Предложение 2.2 *Если среди элементов последовательности $\{b_i\}_{i=1}^{\infty}$ только два различных и оба они имеют бесконечную краность, тогда любая мартингальная мера $P \in \mathcal{P}(Z, \mathbf{F})$ удовлетворяет ОУНБ.*

Доказательство. Не нарушая общности, будем считать, что

$$\begin{aligned} b_1 &= b_3 = b_5 = \dots = b' \\ b_2 &= b_4 = b_6 = \dots = b'' \end{aligned}$$

Тогда для $\forall i \in \mathbb{N}$ и для $\forall J : i \notin J$ (\bar{J} -конечно), $\forall P \in \mathcal{P}(Z, \mathbf{F})$ справедливо неравенство:

$$b' < \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j} < b''$$

То есть ОУНБ выполнено. ■

Теорема 2.1 *Пусть среди элементов последовательности $\{b_i\}_{i=1}^{\infty}$ три различных значения: b_1, b_2, b_3 . Тогда если $b_1 < a < b_2 < b_3$, $m_1 = 1, m_2 = m_3 = \infty$, ($b_1 < b_2 < a < b_3$, $m_1 = m_2 = \infty$, $m_3 = 1$), то $\text{ОУНБ}(Z) = \mathcal{P}(Z, \mathbf{F})$.*

Доказательство. Рассмотрим случай, когда $m_1 = 1, m_2 = m_3 = \infty$. Переобозначим элементы последовательности $\{b_i\}_{i=1}^{\infty}$ следующим образом:

$$b_1;$$

$$b_2 = b_4 = \dots = b_{2i} = \dots, \quad 1 \leq i < m_2 + 1;$$

$$b_3 = b_5 = \dots = b_{2i+1} = \dots, \quad 1 \leq i < m_3 + 1.$$

Пусть:

$$x_1 = p_1, \quad x_2 = \sum_{i=1}^{m_2} p_{2i}, \quad x_3 = \sum_{i=1}^{m_3} p_{2i+1}.$$

Тогда систему (1.3) можно записать в виде:

$$\begin{cases} \sum_{k=1}^3 x_k = 1 \\ \sum_{k=1}^3 b_k x_k = a \\ x_k > 0, \quad 1 \leq k \leq 3. \end{cases} \quad (2.1)$$

Система (2.1) совместна, так как выполнено неравенство $a \neq b_k, 1 \leq k \leq 3$. Пусть (x_1, x_2, x_3) — произвольное решение системы (2.1), тогда можно получить мартингальную меру $P = (p_1, p_2, \dots)$, соответствующую этому решению. Проверим, что $P \in \text{ОУНБ}(Z)$:

1) пусть $i = 1$, тогда

$$b_i = b_1 < \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j},$$

неравенство верно. 2) пусть $i = 2j + 1, 1 \leq j < m_3 + 1$, тогда

$$b_i = b_3 > \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j},$$

неравенство верно. 3) пусть $i = 2j, 1 \leq j < m_2 + 1$, причём $1 \in J$. Тогда:

$$b_i = b_2 \neq \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j} = \frac{a - \sum_{j \in \bar{J}} b_j p_j}{1 - \sum_{j \in \bar{J}} p_j} < \frac{a - a \sum_{j \in \bar{J}} p_j}{1 - \sum_{j \in \bar{J}} p_j} = a,$$

неравенство верно. Если $1 \notin J$, тогда неравенство $b_i = b_2 < \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j}$ выполняется автоматически. Таким образом, любая мартингальная мера, которая является решением системы (2.1), удовлетворяет ОУНБ.

Доказательство теоремы в случае, когда $b_1 < b_2 < a < b_3$, $m_1 = m_2 = \infty$, $m_3 = 1$ повторяется дословно, если от процесса Z перейти к процессу $-Z$. ■

В случаях, когда условия предложений 2.1, 2.2 и теоремы 2.1 не выполнены, можно утверждать, что $\text{ОУНБ}(Z)$ непусто и строго вложено в $\mathcal{P}(Z, \mathbf{F})$. В следующих параграфах представлены достаточные условия существования мартингалльных мер, удовлетворяющих ОУНБ.

2.2 ОУНБ в случае, когда последовательность $\{b_i\}_{i=1}^\infty$ содержит r различных значений ($3 \leq r < \infty$)

Не нарушая общности, можно считать, что элементы последовательности $\{b_i\}_{i=1}^\infty$ удовлетворяют неравенству $b_1 < b_2 < \dots < b_r$. Пусть они имеют кратности m_1, m_2, \dots, m_r соответственно. Принимая во внимание сказанное выше, будем считать в дальнейшем, что не менее чем два из этих чисел бесконечны. Изменив нумерацию, легко получаем следующую систему обозначений:

$$\begin{aligned}
 & b_1 = b_{r+1} = b_{2r+1} = \dots = b_{r(j-1)+1} = \dots, 1 \leq j < m_1 + 1, \\
 & \dots\dots\dots \\
 & b_k = b_{r+k} = b_{2r+k} = \dots = b_{r(j-1)+k} = \dots, 1 \leq j < m_k + 1, \\
 & \dots\dots\dots \\
 & b_r = b_{2r} = b_{3r} = \dots = b_{rj} = \dots, 1 \leq j < m_r + 1.
 \end{aligned}$$

Тогда система (1.3) принимает вид:

$$\begin{cases}
 \sum_{k=1}^r \sum_{j=1}^{m_k} p_{r(j-1)+k} = 1 \\
 \sum_{k=1}^r b_k \sum_{j=1}^{m_k} p_{r(j-1)+k} = a \\
 p_{r(j-1)+k} > 0, 1 \leq k \leq r, 1 \leq j < m_k + 1.
 \end{cases} \tag{2.2}$$

Введём обозначение:

$$x_k = \sum_{j=1}^{m_k} p_{r(j-1)+k}. \quad (2.3)$$

С учётом этого запишем систему (2.2) в виде:

$$\begin{cases} \sum_{k=1}^r x_k = 1 \\ \sum_{k=1}^r b_k x_k = a \\ x_k > 0, \quad 1 \leq k \leq r. \end{cases} \quad (2.4)$$

Решение системы (2.2) будем искать следующим образом:

- 1) получим решение (x_1, \dots, x_r) системы (2.4);
- 2) запишем $\forall k$ ($1 \leq k \leq r$) произвольное представление (2.3). Таким образом можно получить любое решение системы (2.2).

В дальнейшем нам понадобится следующая лемма.

Лемма 2.1 Пусть $b_1 < b_2 < \dots < b_r$, $b_1 < a < b_r$, $a \neq b_2, \dots, a \neq b_{r-1}$. Пусть l и n ($1 \leq l < n \leq r$) — два произвольных индекса. Тогда если числа b_1, \dots, b_r и a рациональны и $b_l < a < b_n$, то существуют решения системы (2.4) (x_1, x_2, \dots, x_r) следующих двух типов:

- 1) все компоненты решения рациональны, а числа x_k ($k \neq l, n$) сколь угодно малы;
- 2) компоненты x_l, x_n и x_s иррациональны (s — произвольный фиксированный индекс, $s \neq l, n$), остальные компоненты решения рациональны, числа x_k ($k \neq l, n$) сколь угодно малы.

Доказательство. 1) Выразим из системы (2.4) значения x_l и x_n .

$$\left\{ \begin{array}{l} (b_1 - b_l)x_l + \dots + (b_{l-1} - b_l)x_{l-1} + (b_{l+1} - b_l)x_{l+1} + \dots + \\ + (b_n - b_l)x_n + \dots + (b_r - b_l)x_r = a - b_l; \\ (b_n - b_1)x_l + \dots + (b_n - b_l)x_l + \dots + (b_n - b_{n-1})x_{n-1} + \dots + \\ + (b_n - b_{n+1})x_{n+1} + \dots + (b_n - b_r)x_r = b_n - a; \\ x_j > 0, \quad 1 \leq k \leq r. \end{array} \right.$$

Тогда получим:

$$\left\{ \begin{array}{l} x_n = \frac{a - b_l}{b_n - b_l} + \frac{b_l - b_1}{b_n - b_l}x_1 + \dots + \frac{b_l - b_{l-1}}{b_n - b_l}x_{l-1} - \frac{b_{l+1} - b_l}{b_n - b_l}x_{l+1} - \dots - \\ - \frac{b_{n-1} - b_l}{b_n - b_l}x_{n-1} - \frac{b_{n+1} - b_l}{b_n - b_l}x_{n+1} - \dots - \frac{b_r - b_l}{b_n - b_l}x_r; \\ x_l = \frac{b_n - a}{b_n - b_l} - \frac{b_n - b_1}{b_n - b_l}x_1 - \dots + \frac{b_n - b_{l-1}}{b_n - b_l}x_{l-1} - \frac{b_n - b_{l+1}}{b_n - b_l}x_{l+1} - \\ - \frac{b_n - b_{n-1}}{b_n - b_l}x_{n-1} + \dots + \frac{b_{n+1} - b_n}{b_n - b_l}x_{n+1} + \dots + \frac{b_r - b_n}{b_n - b_l}x_r; \\ x_j > 0, \quad 1 \leq k \leq r. \end{array} \right.$$

Если все x_j , кроме x_l и x_n приравнять к нулю, то получим:

$$\left\{ \begin{array}{l} x_n = \frac{a - b_l}{b_n - b_l} > 0, \\ x_l = \frac{b_n - a}{b_n - b_l} > 0. \end{array} \right.$$

Если выбрать $x_j > 0$ ($j \neq l, j \neq n$) сколь угодно малыми рациональными, то получим требуемое положительное решение системы (2.4).

2) Пусть $b_l < a < b_n$, зафиксируем произвольный индекс $s : 1 \leq s \leq r$, $s \neq l, n$. Без ограничения общности, можем считать, что $s > n$. Тогда:

$$\left\{ \begin{array}{l} x_n = \frac{a - b_l}{b_n - b_l} + \frac{b_l - b_1}{b_n - b_l} x_1 + \dots + \frac{b_l - b_{l-1}}{b_n - b_l} x_{l-1} - \frac{b_{l+1} - b_l}{b_n - b_l} x_{l+1} - \dots - \\ - \frac{b_{n-1} - b_l}{b_n - b_l} x_{n-1} - \frac{b_{n+1} - b_l}{b_n - b_l} x_{n+1} - \dots - \frac{b_s - b_l}{b_n - b_l} x_s - \dots - \frac{b_r - b_l}{b_n - b_l} x_r; \\ x_l = \frac{b_n - a}{b_n - b_l} - \frac{b_n - b_1}{b_n - b_l} x_1 - \dots + \frac{b_n - b_{l-1}}{b_n - b_l} x_{l-1} - \frac{b_n - b_{l+1}}{b_n - b_l} x_{l+1} - \\ - \frac{b_n - b_{n-1}}{b_n - b_l} x_{n-1} + \dots + \frac{b_{n+1} - b_n}{b_n - b_l} x_{n+1} + \dots + \frac{b_s - b_n}{b_n - b_l} x_s + \dots + \frac{b_r - b_n}{b_n - b_l} x_r; \\ x_j > 0, 1 \leq k \leq r. \end{array} \right.$$

Если теперь выбрать $x_j > 0$ ($j \neq l, n$) сколь угодно малыми рациональными, а $x_s > 0$ сколь угодно малым иррациональным, то x_l и x_n тоже будут иррациональными. Таким образом, мы получим требуемое решение системы (2.4). ■

Пусть J — произвольное множество индексов, \bar{J} — конечное дополнение множества J . Обозначим через:

$$J_k = \bar{J} \cap \{r(j-1) + k, 1 \leq j < m_k + 1\}, \forall k (1 \leq k \leq r). \quad (2.5)$$

Так как \bar{J} — конечно, то лишь конечное число множеств J_k — непусто.

Лемма 2.2 *Мартингальная мера P удовлетворяет ОУНБ тогда и только тогда, когда $\forall i$ ($2 \leq i \leq r-1$) выполняются неравенства*

$$\sum_{k=1}^r (b_i - b_k) \sum_{j \in J_k} p_j \neq b_i - a. \quad (2.6)$$

где J_k — произвольные конечные множества, удовлетворяющие (2.5) (множества J_k могут быть пустыми).

Доказательство. Пусть P — мартингальная мера, удовлетворяющая ОУНБ. Тогда для $\forall i \in \mathbb{N}$, $\forall J \subset \mathbb{N}$ такого, что $i \notin J$ (\bar{J} — конечное множество):

$$b_i \neq \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j} = \frac{a - \sum_{k=1}^r b_k \sum_{j \in J_k} p_j}{1 - \sum_{k=1}^r \sum_{j \in J_k} p_j} \iff \sum_{k=1}^r (b_i - b_k) \sum_{j \in J_k} p_j \neq b_i - a.$$

Легко видеть, что при $i = 1$ и $i = r$ последнее неравенство выполняется автоматически. ■

При $r = 3$ общее решение системы (2.4) можно представить в виде:

$$\begin{cases} x_1 = \frac{b_3 - a}{b_3 - b_1} - \frac{b_3 - b_2}{b_3 - b_1} x_2 \\ x_3 = \frac{a - b_1}{b_3 - b_1} - \frac{b_2 - b_1}{b_3 - b_1} x_2, \end{cases} \quad (2.7)$$

где $0 < x_2 < \frac{a - b_1}{b_2 - b_1}$, если $b_1 < a < b_2$, и $0 < x_2 < \frac{b_3 - a}{b_3 - b_2}$, если $b_2 < a < b_3$. Обозначим через: $x_1 = \sum_{j=1}^{m_1} p_{3j-2}$, $x_2 = \sum_{j=1}^{m_2} p_{3j-1}$, $x_3 = \sum_{j=1}^{m_3} p_{3j}$. Для любого фиксированного решения (x_1, x_2, x_3) системы (2.7) по указанной ранее схеме можно получить общее решение системы (2.2).

Согласно лемме 2.2 мартингальная мера $P \in \text{ОУНБ}(Z)$ тогда и только тогда, когда выполняются неравенства

$$(b_2 - b_1) \sum_{J_1} p_j - (b_3 - b_2) \sum_{J_3} p_j \neq b_2 - a, \quad (2.8)$$

где $J_1 \subset \{3j - 2, 1 \leq j < m_1 + 1\}$ и $J_3 \subset \{3j, 1 \leq j < m_3 + 1\}$ — произвольные конечные множества.

Теорема 2.2 Пусть $b_1 < a < b_2 < b_3$, $1 < m_1 \leq \infty$, $1 \leq m_2 \leq \infty$, $1 \leq m_3 \leq \infty$, (либо $b_1 < b_2 < a < b_3$, $1 \leq m_1 \leq \infty$, $1 \leq m_2 \leq \infty$, $1 < m_3 \leq \infty$). Тогда $\text{ОУНБ}(Z) \neq \emptyset$.

Доказательство. Сначала построим некоторую невырожденную мартингальную меру P , а затем докажем, что она удовлетворяет ОУНБ. Докажем, что существует решение (2.7) такое, что

$$x_1 > \frac{b_2 - a}{b_2 - b_1} \quad (2.9)$$

$$x_3 > \frac{a - b_2}{b_3 - b_2}. \quad (2.10)$$

Если $b_1 < a < b_2 < b_3$, то непосредственно проверяется, что $\frac{b_3 - a}{b_3 - b_1} > \frac{b_2 - a}{b_2 - b_1}$. Из первого уравнения системы (2.7) вытекает, что при достаточно малых x_2 выполняется неравенство (2.9), а неравенство (2.10) выполняется тривиально. Точно так же, если $b_1 < b_2 < a < b_3$, то легко видеть, что $\frac{a - b_1}{b_3 - b_1} > \frac{a - b_2}{b_3 - b_2}$. Из второго уравнения системы (2.7) вытекает, что при достаточно малых x_2 выполняется неравенство (2.10), а неравенство (2.9) выполняется тривиально. Таким образом, нужное решение (2.7) с нужными свойствами существует. Обозначим через (x_1, x_2, x_3) одно из таких решений, зафиксируем его.

Пусть константы $C_1 > 0$ и $C_3 > 0$ такие, что $x_1 > \frac{C_1}{b_2 - b_1} > \frac{b_2 - a}{b_2 - b_1}$, и $x_3 > \frac{C_3}{b_3 - b_2} > \frac{a - b_2}{b_3 - b_2}$. Найдём такие $\delta_1 > 0$ и $\delta_3 > 0$, что выполняются неравенства:

$$x_1 - \delta_1 > \frac{C_1}{b_2 - b_1} > \frac{b_2 - a}{b_2 - b_1}, \quad (2.11)$$

$$\delta_1 < \frac{|b_2 - a|}{b_2 - b_1}, \quad (2.12)$$

$$x_3 - \delta_3 > \frac{C_3}{b_3 - b_2} > \frac{a - b_2}{b_3 - b_2}, \quad (2.13)$$

$$\delta_3 < \frac{|a - b_2|}{b_3 - b_2}, \quad (2.14)$$

$$(b_2 - b_1)\delta_1 + (b_3 - b_2)\delta_3 < |b_2 - a|, \quad (2.15)$$

$$\delta_1 < \frac{C_3 - (a - b_2)}{b_2 - b_1}, \quad (2.16)$$

$$\delta_3 < \frac{C_1 - (b_2 - a)}{b_3 - b_2}. \quad (2.17)$$

Если $\frac{b_2 - b_1}{b_3 - b_2}$ иррационально, то выбираем δ_1 и δ_3 рациональными. Если $\frac{b_2 - b_1}{b_3 - b_2}$ рационально, то выбираем δ_1 рациональным, а δ_3 — иррациональным.

Полагаем $p_1 = x_1 - \delta_1$, а (при $j \geq 2$) $p_{3j-2} = \delta_1 r_j^{(1)}$, где $r_j^{(1)}$ — произвольные положительные рациональные числа такие, что $\sum_{j=1}^{m_1} r_j^{(1)} = 1$. Точно так же полагаем $p_3 = x_3 - \delta_3$, а (при $j \geq 2$) $p_{3j} = \delta_3 r_j^{(3)}$, где $r_j^{(3)}$ — произвольные положительные рациональные числа такие, что $\sum_{j=1}^{m_3} r_j^{(3)} = 1$. Пусть при $j \geq 1$ положительные числа p_{3j-1} удовлетворяют условию $\sum_{j=1}^{m_2} p_{3j-1} = 1$.

Итак, мы получили невырожденную вероятностную меру P . Покажем, что она удовлетворяет неравенствам (2.8).

Случай 1: $J_1 = \emptyset, J_3 = \emptyset$. Неравенство (2.8) очевидно.

Случай 2: $J_1 \neq \emptyset, J_3 = \emptyset$. Неравенство (2.8) перепишем в виде:

$$(b_2 - b_1) \sum_{J_1} p_j \neq b_2 - a \iff \sum_{J_1} p_j \neq \frac{b_2 - a}{b_2 - b_1}. \quad (2.18)$$

Если $b_2 < a < b_3$, то неравенство (2.18) очевидно. Пусть $b_1 < a < b_2$. Если $1 \in J_1$, то из (2.11) следует $\sum_{J_1} p_j \geq x_1 - \delta_1 > \frac{b_2 - a}{b_2 - b_1}$, а если $1 \notin J_1$, то из (2.12) вытекает $\sum_{J_1} p_j \leq \delta_1 < \frac{b_2 - a}{b_2 - b_1}$.

Случай 3: $J_1 = \emptyset, J_3 \neq \emptyset$. Рассуждения проводятся, как в случае 2, но с использованием неравенств (2.13) и (2.14).

Случай 4: $J_1 \neq \emptyset, J_3 \neq \emptyset$. Если $1 \notin J_1, 3 \notin J_3$, то

$$|(b_2 - b_1) \sum_{J_1} p_j - (b_3 - b_2) \sum_{J_3} p_j| \leq (b_2 - b_1)\delta_1 + (b_3 - b_2)\delta_3 < |b_2 - a|,$$

что следует из неравенства (2.15). Если $1 \in J_1, 3 \notin J_3$, то, используя неравенства (2.11) и (2.17), получаем:

$$\begin{aligned} & (b_2 - b_1) \sum_{J_1} p_j - (b_3 - b_2) \sum_{J_3} p_j = \\ & = (b_2 - b_1) \left(x_1 - \delta_1 + \sum_{J_1 \setminus \{1\}} p_j \right) - (b_3 - b_2) \sum_{J_3} p_j \geq \end{aligned}$$

$$\begin{aligned}
&\geq (b_2 - b_1)(x_1 - \delta_1) - (b_3 - b_2)\delta_3 > (b_2 - b_1)\frac{C_1}{b_2 - b_1} - (b_3 - b_2) \cdot \frac{C_1 - (b_2 - a)}{b_3 - b_2} = \\
&= C_1 - C_1 + (b_2 - a) = b_2 - a.
\end{aligned}$$

Если $1 \notin J_1$, $3 \in J_3$, то

$$\begin{aligned}
&(b_2 - b_1) \sum_{J_1} p_j - (b_3 - b_2) \sum_{J_3} p_j = \\
&= (b_2 - b_1) \sum_{J_1} p_j - (b_3 - b_2)(x_3 - \delta_3 + \sum_{J_3 \setminus \{3\}} p_j) \leq \\
&\leq \delta_1 - (b_3 - b_2)(x_3 - \delta_3) < (b_2 - b_1)\frac{C_3 - (a - b_2)}{b_2 - b_1} - (b_3 - b_2) \cdot \frac{C_3}{b_3 - b_2} = \\
&= C_3 - (a - b_2) - C_3 = b_2 - a
\end{aligned}$$

(здесь использовались неравенства (2.13) и (2.16)).

Пусть, наконец, $1 \in J_1$, $3 \in J_3$. Легко видеть, что

$$\begin{aligned}
&(b_2 - b_1) \sum_{J_1} p_j - (b_3 - b_2) \sum_{J_3} p_j = b_2 - a \iff \\
&\iff (b_2 - b_1)(x_1 - \delta'_1) - (b_3 - b_2)(x_3 - \delta'_3) \neq b_2 - a \iff \\
&\iff -(b_2 - b_1)\delta'_1 + (b_3 - b_2)\delta'_3 \neq 0,
\end{aligned}$$

где $\delta'_1 \geq 0$ и $\delta'_3 \geq 0$. Одновременное равенство нулю δ'_1 и δ'_3 невозможно: это бы означало, что $m_1 < \infty$ и $m_3 < \infty$, что, в свою очередь, противоречит предположению о наличии среди чисел m_1 , m_2 и m_3 по крайней мере двух бесконечностей. Если одно из чисел δ'_1 и δ'_3 равно нулю, а другое строго больше нуля, то (2.8) выполняется тривиально. Остается рассмотреть ситуацию, когда $\delta'_1 > 0$ и $\delta'_3 > 0$. В этом случае (2.8) выполняется тогда и только тогда, когда

$$\frac{\delta'_3}{\delta'_1} \neq \frac{b_2 - b_1}{b_3 - b_2}. \quad (2.19)$$

Итак, имеем:

$$(b_2 - b_1) \sum_{J_1} p_j - (b_3 - b_2) \sum_{J_3} p_j =$$

$$\begin{aligned}
&= (b_2 - b_1) \left(x_1 - \delta_1 + \sum_{J_1 \setminus \{1\}} p_j \right) - (b_3 - b_2) \left(x_3 - \delta_3 + \sum_{J_3 \setminus \{3\}} p_j \right) = \\
&= (b_2 - b_1) \left(x_1 - \delta_1 \left(1 - \sum_{J_1 \setminus \{1\}} r_j^{(1)} \right) \right) - (b_3 - b_2) \left(x_3 - \delta_3 \left(1 - \sum_{J_3 \setminus \{3\}} r_j^{(3)} \right) \right).
\end{aligned}$$

Обозначим $\delta'_1 = \delta_1 \left(1 - \sum_{J_1 \setminus \{1\}} r_j^{(1)} \right)$, $\delta'_3 = \delta_3 \left(1 - \sum_{J_3 \setminus \{3\}} r_j^{(3)} \right)$. Имеем:

$$\frac{\delta'_3}{\delta'_1} = \frac{\delta_3 \left(1 - \sum_{J_3 \setminus \{3\}} r_j^{(3)} \right)}{\delta_1 \left(1 - \sum_{J_1 \setminus \{1\}} r_j^{(1)} \right)}.$$

Так как выражения в скобках в числителе и знаменателе рациональны, то, принимая во внимание выбор чисел δ_1 и δ_3 , описанный в шаге 4, получаем неравенство $\frac{\delta'_3}{\delta'_1} \neq \frac{b_2 - b_1}{b_3 - b_2}$, что доказывает выполнение (2.8) и в этом случае. ■

Для того чтобы получить достаточные условия существования мартингальных мер, удовлетворяющих ОУНБ в случае $3 < r < \infty$, нам понадобится следующий факт.

Замечание 2.2 Пусть $0 < x_k < 1$ ($m_k = \infty$, $1 \leq k \leq r$) и $d > 1$ — натуральное число. В доказательстве следующей теоремы мы будем использовать d -адическое представление вида: $x_k = \sum_{i=1}^{\infty} \frac{t_i^{(k)}}{d^i}$, где натуральные числа $t_i^{(k)}$ могут принимать значение $0, 1, 2, \dots, d-1$. Хорошо известно, что для любого такого x_k существует представление указанного вида, в котором ряд $\sum_{i=1}^{\infty} \frac{t_i^{(k)}}{d^i}$ содержит бесконечное число строго положительных членов. Именно такие представления мы будем всегда рассматривать. В качестве компонент $p_{r(j-1)+k}$ ($1 \leq j < \infty$) мы будем выбирать j -й ненулевой член ряда $\sum_{i=1}^{\infty} \frac{t_i^{(k)}}{d^i}$.

Теорема 2.3 Пусть числа $b_1 < b_2 < \dots < b_r$ рациональны, a — действительное число, $b_1 < a < b_r$, $a \neq b_k$ ($k = 2, 3, \dots, r-1$). Тогда $\text{ОУНБ}(Z) \neq \emptyset$.

Доказательство. Сначала рассмотрим случай, когда число a иррационально. Ясно, что существуют индексы l, n ($1 \leq l < n \leq r$) такие, что $m_l = m_n = \infty$. Зафиксируем решение (x_1, x_2, \dots, x_r) системы (2.4) из первой части леммы 2.1 (координаты x_k , $k \neq l, n$ рациональны).

Если k таково, что $m_k = \infty$, то представляем x_k в виде $x_k = \sum_{i=1}^{\infty} \frac{t_i^{(k)}}{2^i}$, где $t_i^{(k)} \in \{0, 1\}$. Определяем $p_{r(j-1)+k}$ ($1 \leq j < \infty$), как это описано в замечании 2.2 (таким образом, все эти числа рациональны). Если k таково, что $m_k < \infty$ (в этом случае $k \neq l, n$), то полагаем $p_{r(j-1)+k} = \frac{x_k}{m_k}$ ($1 \leq j \leq m_k$). Так как координаты x_k рациональны при $m_k < \infty$, то и соответствующие числа $p_{r(j-1)+k}$ рациональны.

Итак, мы построили невырожденную мартингальную меру P . Покажем, что $P \in \text{ОУНБ}(Z)$.

Так как в (2.6) все множества J_k ($1 \leq k \leq r$) конечны, то ясно, что левые части этих соотношений рациональны, в то время как правые — иррациональны. Таким образом, неравенства (2.6) выполняются, то есть построенная мера P удовлетворяет ОУНБ и теорема 2.3 в случае иррационального a доказана.

Теперь рассмотрим случай, когда число a рационально. Принимая во внимание замечание 2.1, можно считать, что числа b_k , $k = 1, 2, \dots, r$ и a — целые.

1) Предположим, что существуют такие индексы l и n ($1 \leq l < n \leq r$), что $m_l = m_n = \infty$ и $b_l < a < b_n$. Зафиксируем решение (x_1, x_2, \dots, x_r) системы (2.4) из второй части леммы 2.1, являющееся решением 1-го

типа. Так как

$$\sum_{\substack{k=1, \\ k \neq l, n}}^r |b_i - b_k| \sum_{J_k} p_j \leq \sum_{\substack{k=1, \\ k \neq l, n}}^r |b_i - b_k| x_k,$$

то будем считать компоненты x_k ($k \neq l, n$) настолько малыми, что для всех i ($1 \leq i \leq r$) выполняются неравенства:

$$\left| \sum_{\substack{k=1, \\ k \neq l, n}}^r (b_i - b_k) \sum_{J_k} p_j \right| < |b_i - a|. \quad (2.20)$$

Для индексов k ($k \neq l, n$) поступим так же, как в предыдущей части доказательства. Следовательно, для индексов k таких, что $m_k = \infty$, $\sum_{J_k} p_j = \frac{C_k}{2^{\gamma_k}}$, где C_k, γ_k — некоторые неотрицательные целые числа ($k \neq l, n$, $0 \leq C_k < 2^{\gamma_k}$, C_k и γ_k зависят от J_k). Для индексов k таких, что $m_k < \infty$, представим $x_k = \frac{C'_k}{C''_k}$, где C'_k, C''_k — положительные целые числа.

Тогда $\sum_{J_k} p_j = \frac{|J_k| C'_k}{C''_k \cdot m_k}$, где $|J_k|$ — число элементов множества J_k . Зададим простое число $d_l > b_r - b_1$ настолько большим, что $d_l > \max_{\substack{1 \leq k \leq r, \\ m_k < \infty}} \{C''_k + m_k\}$,

и простое число $d_n > d_l$. Теперь представляем $x_l = \sum_{i=1}^{\infty} \frac{t_i^{(l)}}{d_l^i}$, $x_n = \sum_{i=1}^{\infty} \frac{t_i^{(n)}}{d_n^i}$ и задаём $p_{r(j-1)+l}$, $p_{r(j-1)+n}$ ($1 \leq j < \infty$), как описано в замечании 2.2.

Ясно, что

$$\sum_{J_l} p_j = \frac{C_l}{d_l^{\gamma_l}} \quad (0 \leq C_l < d_l^{\gamma_l}, C_l \text{ и } \gamma_l \text{ зависят от } J_l),$$

$$\sum_{J_n} p_j = \frac{C_n}{d_n^{\gamma_n}} \quad (0 \leq C_n < d_n^{\gamma_n}, C_n \text{ и } \gamma_n \text{ зависят от } J_n).$$

Итак, мы построили невырожденную мартингальную меру P . Проверим выполнение для нее неравенств (2.6).

Если $J_l = \emptyset$, $J_n = \emptyset$, то (2.6) вытекают из неравенств (2.20).

Если $J_l \neq \emptyset$, $J_n = \emptyset$, то (2.6) равносильны неравенствам

$$\frac{(b_i - b_l) \cdot C_l}{d_l^{\gamma_l}} \neq b_i - a - \sum_{\substack{k=1, \\ k \neq l, n}}^r (b_i - b_k) \sum_{J_k} p_j. \quad (2.21)$$

Пусть $i \neq l$. Из условий на d_l вытекает, что в левой части стоит несократимая рациональная дробь. Если в правой части произвести алгебраические действия и полученную рациональную дробь сократить, то в несократимой дроби в знаменателе не может содержаться сомножитель d_l . Поэтому неравенства (2.21), а следовательно, и (2.6) выполняются. Если $i = l$, то (2.21) следует из (2.20).

Случай $J_l = \emptyset, J_n \neq \emptyset$ рассматриваются аналогично предыдущему.

Пусть $J_l \neq \emptyset, J_n \neq \emptyset$. Неравенства (2.6) равносильны неравенствам:

$$\frac{(b_i - b_n) \cdot C_n}{d_l^n} \neq b_i - a - \frac{(b_i - b_l)C_l}{d_l^n} - \sum_{\substack{k=1, \\ k \neq l, n}}^r (b_i - b_k) \sum_{J_k} p_j. \quad (2.22)$$

Рассуждая так же, как и в случае $J_l \neq \emptyset, J_n = \emptyset$, получаем, что (2.22) выполнено.

Таким образом, построенная мартингальная мера P удовлетворяет ОУНБ.

2) Если не выполняется предположение 1) в предыдущей части доказательства, то существуют такие индексы l и n , что $b_l < a < b_n$ и среди чисел m_l и m_n одно бесконечно, а другое — конечно. Не нарушая общности, можно считать, что $m_l = \infty$, а $m_n < \infty$.

Зафиксируем такое решение 2-го типа системы (2.4) из второй части леммы 2.1, что выполняется неравенство (2.20). Для индексов k ($k \neq l, n$) поступим так же, как в предыдущей части доказательства. Так же, как и ранее, выбираем простое число d_l , представляем $x_l = \sum_{i=1}^{\infty} \frac{t_i^{(l)}}{d_l^i}$ и задаем $p_{r(j-1)+l}$. Наконец, полагаем $p_{r(j-1)+n} = \frac{x_n}{m_n}$ ($1 \leq j \leq m_n$). Покажем, что полученная невырожденная мартингальная мера P удовлетворяет неравенствам (2.6).

Случаи $J_l = \emptyset, J_n = \emptyset$ и $J_l \neq \emptyset, J_n = \emptyset$ рассматриваются так же, как и в предыдущей части доказательства. Пусть $J_l = \emptyset, J_n \neq \emptyset$. Тогда

(2.6) равносильно неравенствам

$$\frac{(b_i - b_n) \cdot |J_n| \cdot x_n}{m_n} \neq b_i - a - \sum_{\substack{k=1, \\ k \neq l, n}}^r (b_i - b_k) \sum_{J_k} p_j. \quad (2.23)$$

При $i \neq n$ это неравенство выполняется, так как число, стоящее в левой части, иррационально, а в правой — рационально. При $i = n$ (2.23) вытекает из (2.20).

Пусть $J_l \neq \emptyset$, $J_n \neq \emptyset$. Неравенства (2.6) равносильны неравенствам:

$$\frac{(b_i - b_n) \cdot |J_n| \cdot x_n}{m_n} \neq b_i - a - \frac{(b_i - b_l)C_l}{d_l^n} - \sum_{\substack{k=1, \\ k \neq l, n}}^r (b_i - b_k) \sum_{J_k} p_j. \quad (2.24)$$

Рассуждая, как и ранее, показываем, что неравенства (2.24) справедливы. Таким образом, построенная мартингальная мера P удовлетворяет ОУНБ. ■

Следует отметить, что в рассматриваемых случаях существуют мартингальные меры, не удовлетворяющие ОУНБ (см.[26]), поэтому ОУНБ(Z) строго вложено в $\mathcal{P}(Z, \mathbf{F})$.

2.3 Построение мартингальных мер, удовлетворяющих ОУНБ в случае, когда последовательность $\{b_i\}_{i=1}^\infty$ содержит 3 различных значения

В данном параграфе будет предложен конструктивный способ построения мартингальных мер, удовлетворяющих ОУНБ. На основе полученных формул, можно создать математические модели безарбитражных неполных финансовых рынков, которые с помощью метода хааровских интерполяций преобразуются в полные. Эти модели и сопровождающие их расчёты будут представлены в программном комплексе, которому будет посвящена глава 3.

Рассмотрим случай, когда среди элементов последовательности $\{b_i\}_{i=1}^{\infty}$ три различных значения: b_1, b_2, b_3 . Пусть они имеют кратности m_1, m_2, m_3 , соответственно. Изменив нумерацию элементов, получим следующую систему обозначений:

$$b_1 = b_4 = b_7 = \dots = b_{3j-2} = \dots, \quad 1 \leq j < m_1 + 1,$$

$$b_2 = b_5 = b_8 = \dots = b_{3j-1} = \dots, \quad 1 \leq j < m_2 + 1,$$

$$b_3 = b_6 = b_9 = \dots = b_{3j} = \dots, \quad 1 \leq j < m_3 + 1,$$

причём не менее двух цепочек этих равенств должны быть бесконечными. Если, например, $m_1 = 3 < \infty$, то первая цепочка конечна и имеет вид: $b_1 = b_4 = b_7$, то индексы $9, 12, 15, \dots$ будут во вновь принятых обозначениях отсутствовать. Без ограничения общности, будем считать, что $b_1 < a < b_2 < b_3$. Рассмотрим систему:

$$\begin{cases} \sum_{j=1}^{m_1} p_{3j-2} + \sum_{j=1}^{m_2} p_{3j-1} + \sum_{j=1}^{m_3} p_{3j} = 1 \\ b_1 \sum_{j=1}^{m_1} p_{3j-2} + b_2 \sum_{j=1}^{m_2} p_{3j-1} + b_3 \sum_{j=1}^{m_3} p_{3j} = a \\ p_j > 0, \quad j \in \mathbb{N}. \end{cases} \quad (2.25)$$

Пусть J — произвольное множество из определения ОУНБ, а \bar{J} — его конечное дополнение. Представ \bar{J} в виде:

$$\bar{J} = J_1 \cup J_2 \cup J_3,$$

где $J_1 \subset \{3j - 2, j < m_1 + 1\}$, $J_2 \subset \{3j - 1, j < m_2 + 1\}$, $J_3 \subset \{3j, j < m_3 + 1\}$.

Замечание 2.3 Если $i = 3j - 2$, $j < m_1 + 1$, или $i = 3j$, $j < m_3 + 1$, то ОУНБ выполняется автоматически.

Лемма 2.3 Пусть $i \in \{3j - 1, j < m_2 + 1\}, J_1 \subset \{3j - 2, j < m_1 + 1\}, J_3 \subset \{3j, j < m_3 + 1\}$. Тогда ОУНБ равносильно неравенству:

$$(b_2 - b_1) \left(\sum_{j=1}^{m_1} p_{3j-2} - \sum_{J_1} p_j \right) \neq (b_3 - b_2) \left(\sum_{j=1}^{m_3} p_{3j} - \sum_{J_3} p_j \right) \quad (2.26)$$

при любом $J_2 \subset \{3j - 1, j < m_2 + 1\}$.

Доказательство. Имеем:

$$\begin{aligned} b_{3i-1} = b_2 &\neq \frac{\sum_J b_j p_j}{\sum_J p_j} = \\ &= \frac{\sum_{j=1}^{m_1} b_{3j-2} p_{3j-2} - \sum_{J_1} b_j p_j + \sum_{j=1}^{m_2} b_{3j-1} p_{3j-1} - \sum_{J_2} b_j p_j + \sum_{j=1}^{m_3} b_{3j} p_{3j} - \sum_{J_3} b_j p_j}{\sum_{j=1}^{m_1} p_{3j-2} - \sum_{J_1} p_j + \sum_{j=1}^{m_2} p_{3j-1} - \sum_{J_2} p_j + \sum_{j=1}^{m_3} p_{3j} - \sum_{J_3} p_j} = \\ &= \frac{b_1 \sum_{j=1}^{m_1} p_{3j-2} - b_1 \sum_{J_1} p_j + b_2 \sum_{j=1}^{m_2} p_{3j-1} - b_2 \sum_{J_2} p_j + b_3 \sum_{j=1}^{m_3} p_{3j} - b_3 \sum_{J_3} p_j}{\sum_{j=1}^{m_1} p_{3j-2} - \sum_{J_1} p_j + \sum_{j=1}^{m_2} p_{3j-1} - \sum_{J_2} p_j + \sum_{j=1}^{m_3} p_{3j} - \sum_{J_3} p_j} \Leftrightarrow \\ &= (b_2 - b_1) \left(\sum_{j=1}^{m_1} p_{3j-2} - \sum_{J_1} p_j \right) \neq (b_3 - b_2) \left(\sum_{j=1}^{m_3} p_{3j} - \sum_{J_3} p_j \right). \end{aligned}$$

■

Построим мартингалные меры, удовлетворяющие ОУНБ, при различных предположениях относительно кратностей m_1, m_2, m_3 .

Положим:

$$p_{3j-2} = \frac{c}{(b_2 - b_1) 2^{n(3j-2)}}, \quad j < m_1 + 1, \quad p_{3j} = \frac{c}{(b_3 - b_2) 2^{n3j}}, \quad j < m_3 + 1, \quad (2.27)$$

где $c > 0$ — некоторая константа, а положительные числа $p_{3j-1}, j < m_2 + 1$ выбираются произвольным образом, но так, чтобы выполнялось равенство:

$$\sum_{j=1}^{m_1} p_{3j-2} + \sum_{j=1}^{m_2} p_{3j-1} + \sum_{j=1}^{m_3} p_{3j} = 1.$$

Обозначим через:

$$d := \sum_{j=1}^{m_2} p_{3j-1}. \quad (2.28)$$

С учётом всевозможных значений m_1, m_2, m_3 можно выделить три случая.

Случай 1. Пусть $m_1 = m_3 = \infty, m_2 \leq \infty$. Возьмём $n \in \mathbb{N}$ настолько большим, что справедливо неравенство:

$$\frac{(b_2 - a)}{(2^{2n} - 1)} \left(\frac{2^{2n}}{b_2 - b_1} + \frac{1}{b_3 - b_2} \right) < 1 \quad (2.29)$$

Учитывая, что

$$\sum_{j=1}^{\infty} \frac{1}{2^{n(3j-2)}} = \frac{2^{2n}}{2^{3n} - 1}, \quad \sum_{j=1}^{\infty} \frac{1}{2^{n3j}} = \frac{1}{2^{3n} - 1},$$

запишем систему (2.25) в форме:

$$\begin{cases} \frac{c}{(b_2-b_1)} \frac{2^{2n}}{(2^{3n}-1)} + d + \frac{c}{(b_3-b_2)} \frac{1}{(2^{3n}-1)} = 1 \\ \frac{b_1c}{(b_2-b_1)} \frac{2^{2n}}{(2^{3n}-1)} + b_2d + \frac{b_3c}{(b_3-b_2)} \frac{1}{(2^{3n}-1)} = a \\ c > 0, d > 0 \end{cases} \quad (2.30)$$

Решая систему (2.30) относительно c и d , получим:

$$\begin{cases} c = \frac{(b_2-a)(2^{3n}-1)}{(2^{2n}-1)} \\ d = 1 - \frac{(b_2-a)}{(b_2-b_1)} \frac{2^{2n}}{(2^{2n}-1)} - \frac{(b_2-a)}{(b_3-b_2)} \frac{1}{(2^{2n}-1)}. \end{cases}$$

Ясно, что $c > 0$, а из неравенства (2.29) вытекает, что $d > 0$.

Убедимся в том, что построенная мартингальная мера удовлетворяет ОУНБ, т.е. для неё выполнено неравенство (2.26):

$$\begin{aligned} (b_2 - b_1) \left(\frac{1}{(b_2 - b_1)} \frac{2^{2n}}{2^{3n} - 1} - \frac{1}{(b_2 - b_1)} \sum_{J_1} \frac{1}{2^{n(3j-2)}} \right) &\neq \\ (b_3 - b_2) \left(\frac{1}{(b_3 - b_2)} \frac{1}{2^{3n} - 1} - \frac{1}{(b_3 - b_2)} \sum_{J_3} \frac{1}{2^{3nj}} \right) &\Leftrightarrow \end{aligned}$$

$$\begin{aligned} \frac{2^{2n} - 1}{2^{3n} - 1} \neq \sum_{J_1} \frac{1}{2^{n(3j-2)}} - \sum_{J_3} \frac{1}{2^{n3j}} &\Leftrightarrow \frac{2^n + 1}{2^{2n} + 2^n + 1} \neq \sum_{J_1} \frac{1}{2^{n(3j-2)}} - \sum_{J_3} \frac{1}{2^{n3j}} \Leftrightarrow \\ &\frac{2^{2n}}{2^{2n} + 2^n + 1} \neq 1 + \sum_{J_3} \frac{1}{2^{n3j}} - \sum_{J_1} \frac{1}{2^{n(3j-2)}}. \end{aligned}$$

Последнее неравенство верно, так как в левой части стоит несократимая неправильная дробь с нечётным знаменателем, а в правой будет либо целое число, либо несократимая дробь с чётным знаменателем.

Таким образом, построенная мартингальная мера P , удовлетворяет ОУНБ.

Случай 2. Пусть $m_1 < \infty$, $m_2 = m_3 = \infty$. Компоненты мартингальной меры P будем определять по формулам (2.27), (2.28).

$$\sum_{j=1}^{m_1} \frac{1}{2^{n(3j-2)}} = \frac{2^{2n}(2^{3nm_1} - 1)}{(2^{3n} - 1)2^{3nm_1}}, \quad \sum_{j=1}^{\infty} \frac{1}{2^{3nj}} = \frac{1}{2^{3n} - 1}.$$

Возьмём $n \in \mathbb{N}$ настолько большим, что справедливо неравенство:

$$\frac{(b_2 - a)(2^{3n} - 1)2^{3nm_1}}{2^{3nm_1}(2^{2n} - 1) - 2^{2n}} \frac{1}{(2^{3n} - 1)} \left(\frac{2^{2n}(2^{3nm_1} - 1)}{2^{3nm_1}(b_2 - b_1)} + \frac{1}{(b_3 - b_2)} \right) < 1 \quad (2.31)$$

Запишем систему (2.25) в виде:

$$\begin{cases} \frac{c}{(b_2 - b_1)} \frac{2^{2n}(2^{3nm_1} - 1)}{(2^{3n} - 1)2^{3nm_1}} + d + \frac{c}{(b_3 - b_2)} \frac{1}{(2^{3n} - 1)} = 1 \\ \frac{b_1 c}{(b_2 - b_1)} \frac{2^{2n}(2^{3nm_1} - 1)}{(2^{3n} - 1)2^{3nm_1}} + b_2 d + \frac{b_3 c}{(b_3 - b_2)} \frac{1}{(2^{3n} - 1)} = a \\ c > 0, \quad d > 0. \end{cases} \quad (2.32)$$

Решая систему (2.32) относительно c и d , получим:

$$\begin{cases} c = \frac{(b_2 - a)(2^{3n} - 1)2^{3nm_1}}{2^{3nm_1}(2^{2n} - 1) - 2^{2n}}; \\ d = 1 - \frac{c}{(2^{3n} - 1)} \left(\frac{2^{2n}(2^{3nm_1} - 1)}{2^{3nm_1}(b_2 - b_1)} - \frac{1}{(b_3 - b_2)} \right). \end{cases}$$

Поскольку $2^{3nm_1}(2^{2n} - 1) - 2^{2n} > 0$, $\forall n \in \mathbb{N}$, $\forall m_1 \geq 1$, то $c > 0$, а из неравенства (2.31) следует, что $d > 0$. Проверим, что для мартингальной меры P , выполнено неравенство (2.26):

$$(b_2 - b_1) \left(\frac{1}{(b_2 - b_1)} \frac{2^{2n}(2^{3nm_1} - 1)}{(2^{3n} - 1)2^{3nm_1}} - \frac{1}{(b_2 - b_1)} \sum_{J_1} \frac{1}{2^{n(3j-2)}} \right) \neq$$

$$\begin{aligned}
 & \neq (b_3 - b_2) \left(\frac{1}{(b_3 - b_2) 2^{3n} - 1} - \frac{1}{(b_3 - b_2)} \sum_{J_3} \frac{1}{2^{3nj}} \right) \Leftrightarrow \\
 & \frac{2^{2n}(2^{3nm_1} - 1)}{(2^{3n} - 1)2^{3nm_1}} - \sum_{J_1} \frac{1}{2^{n(3j-2)}} \neq \frac{1}{2^{3n} - 1} - \sum_{J_3} \frac{1}{2^{3nj}} \Leftrightarrow \\
 & \frac{2^{2n}(2^{3nm_1} - 1)}{(2^{3n} - 1)2^{3nm_1}} - \frac{1}{2^{3n} - 1} \neq \sum_{J_1} \frac{1}{2^{n(3j-2)}} - \sum_{J_3} \frac{1}{2^{3nj}} \Leftrightarrow \\
 & \frac{2^{2n}(2^{3n} - 1)(2^{3n(m_1-1)} + 2^{3n(m_1-2)} + \dots + 2^{3n} + 1)}{(2^{3n} - 1)2^{3nm_1}} - \frac{1}{2^{3n} - 1} \neq \\
 & \neq \sum_{J_1} \frac{1}{2^{n(3j-2)}} - \sum_{J_3} \frac{1}{2^{3nj}} \Leftrightarrow \\
 & \frac{1}{2^{3n} - 1} \neq \frac{2^{3n(m_1-1)} + 2^{3n(m_1-2)} + \dots + 2^{3n} + 1}{2^{n(3m_1-2)}} - \sum_{J_1} \frac{1}{2^{n(3j-2)}} + \sum_{J_3} \frac{1}{2^{3nj}}.
 \end{aligned}$$

Последнее неравенство верно. Поэтому построенная мартингаловая мера P удовлетворяет ОУНБ.

Случай 3. Пусть $m_1 = m_2 = \infty$, $m_3 < \infty$. Компоненты мартингалового процесса P также будем определять по формулам (2.27), (2.28). Учтывая, что:

$$\sum_{j=1}^{\infty} \frac{1}{2^{n(3j-2)}} = \frac{2^{2n}}{2^{3n} - 1}; \quad \sum_{j=1}^{m_3} \frac{1}{2^{3nj}} = \frac{2^{3nm_3} - 1}{2^{3nm_3}(2^{3n} - 1)},$$

возьмём $n \in \mathbb{N}$ настолько большим, что справедливо неравенство:

$$\frac{(b_2 - a)2^n}{(2^{3n} - 1)} \left(\frac{2^{2n}}{(b_2 - b_1)} + \frac{2^{3nm_3} - 1}{(b_3 - b_2)2^{3nm_3}} \right) < 1 \quad (2.33)$$

Запишем систему (2.25) в виде:

$$\begin{cases} \frac{c}{(b_2 - b_1)} \frac{2^{2n}}{(2^{3n} - 1)} + d + \frac{c}{(b_3 - b_2)} \frac{(2^{3nm_3} - 1)}{(2^{3n} - 1)2^{3nm_3}} = 1 \\ \frac{b_1 c}{(b_2 - b_1)} \frac{2^{2n}}{(2^{3n} - 1)} + b_2 d + \frac{b_3 c}{(b_3 - b_2)} \frac{(2^{3nm_3} - 1)}{(2^{3n} - 1)2^{3nm_3}} = a \\ c > 0, d > 0. \end{cases} \quad (2.34)$$

Решая систему (2.34) относительно c и d , получим:

$$\begin{cases} c = \frac{(b_2-a)(2^{3n}-1)2^{3nm_3}}{2^{3nm_3}(2^{2n}-1)+1}; \\ d = 1 - \frac{c}{(2^{3n}-1)} \left(\frac{2^{2n}}{(b_2-b_1)} - \frac{1}{(b_3-b_2)} \frac{2^{3nm_3}-1}{2^{3nm_3}} \right). \end{cases}$$

Учитывая, что $c > 0$ и $c \sim (b_2 - a)$, $n \rightarrow \infty$, получим:

$$d \sim \frac{(b_2 - a)2^n}{2^{3n} - 1} \left(\frac{2^{2n}}{(b_2 - b_1)} - \frac{1}{(b_3 - b_2)} \frac{2^{3nm_3} - 1}{2^{3nm_3}} \right), \quad n \rightarrow \infty.$$

Из неравенства (2.33) следует, что при достаточно большом n , $d > 0$.

Проверим, что для мартингальной меры P , выполнено неравенство (2.26):

$$\begin{aligned} \frac{2^{2n}}{2^{3n} - 1} - \sum_{J_1} \frac{1}{2^{n(3j-2)}} &\neq \frac{2^{3nm_3} - 1}{2^{3nm_3}(2^{3n} - 1)} - \sum_{J_3} \frac{1}{2^{3nj}} \Leftrightarrow \\ \frac{2^{2n}}{2^{3n} - 1} - \frac{2^{3nm_3} - 1}{2^{3nm_3}(2^{3n} - 1)} &\neq \sum_{J_1} \frac{1}{2^{n(3j-2)}} - \sum_{J_3} \frac{1}{2^{3nj}} \Leftrightarrow \\ \frac{2^{2n}}{2^{3n} - 1} - \frac{2^{2n}(2^{3n} - 1)(2^{3n(m_3-1)} + 2^{3n(m_3-2)} + \dots + 2^{3n} + 1)}{(2^{3n} - 1)2^{3nm_3}} &\neq \\ \sum_{J_1} \frac{1}{2^{n(3j-2)}} - \sum_{J_3} \frac{1}{2^{3nj}} &\Leftrightarrow \\ \frac{2^{2n}}{2^{3n} - 1} &\neq \frac{2^{3n(m_3-1)} + 2^{3n(m_3-2)} + \dots + 2^{3n} + 1}{2^{3nm_3}} + \sum_{J_1} \frac{1}{2^{n(3j-2)}} - \sum_{J_3} \frac{1}{2^{3nj}}. \end{aligned}$$

Нетрудно убедиться в том, что последнее неравенство верно. Таким образом, мартингальная мера P удовлетворяет ОУНБ.

Рассмотрим случай, когда: $b_1 < b_2 < a < b_3$. Если перейти к процессу $-Z = (Z_k, \mathcal{F}_k)_{k=0}^1$, тогда будет выполняться неравенство: $-b_3 < -a < -b_2 < -b_1$. Так как $\mathcal{P}(-Z, \mathbf{F}) = \mathcal{P}(Z, \mathbf{F})$ и $\text{ОУНБ}(-Z) = \text{ОУНБ}(Z)$, поэтому из полученного следует, что $\text{ОУНБ}(-Z) \neq \emptyset$. Таким образом, если среди элементов последовательности $\{b_i\}_{i=1}^\infty$ всего три различных значения, то все возможные случаи рассмотрены и для них получены мартингальные меры, удовлетворяющие ОУНБ.

2.4 Множество мартингальных мер, удовлетворяющих ОУНБ в случае, когда последовательность $\{b_i\}_{i=1}^{\infty}$ содержит бесконечное число различных значений

Пусть все значения последовательности $\{b_i\}_{i=1}^{\infty}$ различны, $b_i \in \mathbb{R}$. Не нарушая общности, будем считать, что они монотонно возрастают: $b_1 < b_2 < b_3 < \dots$. Для доказательства следующей теоремы нам понадобится лемма.

Лемма 2.4 Пусть $P \in \mathcal{P}(Z, \mathbf{F})$ — произвольная мартингальная мера и $b_1 < a < b_2 < b_3 < b_4 < \dots$, причём $b_i \geq 2b_{i-1}$, $\forall i \geq 2$. Для того, чтобы мера P удовлетворяла ОУНБ достаточно, чтобы

$$b_{i-1} \min_{2 \leq j \leq i-1} p_j > \sum_{j=i+1}^{\infty} b_j p_j, \quad \forall i \geq 3. \quad (2.35)$$

В частности, если последовательность p_2, p_3, \dots монотонно убывает, то неравенство (2.35) совпадает с неравенством

$$b_{i-1} p_{i-1} > \sum_{j=i+1}^{\infty} b_j p_j, \quad \forall i \geq 3. \quad (2.36)$$

Доказательство. Пусть $i \geq 2$, $J \subset \{2, 3, 4, \dots\} \setminus \{i\}$ (остальные неравенства из ОУНБ выполняются автоматически без применения (2.35)). Представим множество $J = J' \cup J''$, где $J' \subset \{2, 3, 4, \dots, i-1\}$, $J'' \subset \{i+1, i+2, i+3, \dots\}$.

Если $J' = \emptyset$, то ОУНБ

$$\frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j} = \frac{\sum_{j \in J''} b_j p_j}{\sum_{j \in J''} p_j} \geq b_{i+1} > b_i$$

верно. Если $J' \neq \emptyset$, то $\forall i \geq 3$:

$$b_i \neq \frac{\sum_{j \in J} b_j p_j}{\sum_{j \in J} p_j} \iff b_i \sum_{j \in J'} p_j - \sum_{j \in J'} b_j p_j \neq \sum_{j \in J''} b_j p_j - b_i \sum_{j \in J'} p_j.$$

последнее неравенство верно, так как:

$$\begin{aligned} b_i \sum_{j \in J'} p_j - \sum_{j \in J'} b_j p_j &\geq (b_i - b_{i-1}) \sum_{j \in J'} p_j \geq b_{i-1} \min_{2 \leq j \leq i-1} p_j > \sum_{j=i+1}^{\infty} b_j p_j \geq \\ &\sum_{j \in J''} b_j p_j \geq \sum_{j \in J''} b_j p_j - b_i \sum_{j \in J''} p_j. \end{aligned}$$

Таким образом, мартингальная мера P удовлетворяет ОУНБ. ■

Теорема 2.4 Пусть элементы последовательности $\{b_i\}_{i=1}^{\infty}$ удовлетворяют неравенству: $b_1 < a < b_2 < b_3 < b_4 < \dots$, причём

$$b_i \geq 2b_{i-1}, \quad \forall i \geq 2. \quad (2.37)$$

Тогда $OУНБ(Z) \neq \emptyset$ и строго вложено в $\mathcal{P}(Z, \mathbf{F})$.

Доказательство. Не нарушая общности, будем считать, что $b_1 \geq 1$. Покажем сначала, что множество мартингальных мер, для которых выполнено условие (2.35), непусто. Для этого запишем систему (1.3) следующим образом:

$$\begin{cases} p_1 + p_2 + \sum_{j=3}^{\infty} p_j = 1 \\ b_1 p_1 + b_2 p_2 + \sum_{j=3}^{\infty} b_j p_j = a \\ p_i > 0, \quad i \in \mathbb{N}. \end{cases} \quad (2.38)$$

Построим такое решение системы (2.38), что для него выполнено (2.35) и (2.37). Возьмём $p_j = \frac{1}{b_j 2^{nj}}$, $\forall j \geq 3$. При этом число n зададим настолько большим, чтобы выполнялись неравенства: $\frac{a - \varepsilon}{b_1} > 1$, $\frac{a - \varepsilon}{b_2} < 1$, $1 - \delta > \frac{a - \varepsilon}{b_2}$, $\frac{a - b_1}{b_2 - b_1} + \frac{b_1 \delta - \varepsilon}{b_2 - b_1} > \max\{\frac{\varepsilon}{b_2}, \frac{1}{b_3 2^{3n}}\}$, где $\varepsilon = \sum_{j=3}^{\infty} \frac{1}{2^{nj}} = \frac{1}{2^{2n}(2^n - 1)}$, а $\delta = \sum_{j=3}^{\infty} \frac{1}{b_j 2^{nj}} < \varepsilon$. Тогда система (2.38) примет вид:

$$\begin{cases} p_1 + p_2 = 1 - \delta \\ b_1 p_1 + b_2 p_2 = a - \varepsilon \\ p_i > 0, \quad i = 1, 2 \end{cases} \quad (2.39)$$

Легко видеть, что система (2.39) имеет единственное решение:

$$\begin{cases} p_1 = \frac{b_2 - a}{b_2 - b_1} + \frac{\varepsilon + b_2 \delta}{b_2 - b_1} \\ p_2 = \frac{a - b_1}{b_2 - b_1} + \frac{b_1 \delta - \varepsilon}{b_2 - b_1} \end{cases} \quad (2.40)$$

Таким образом, мы получили невырожденную мартингальную меру. Покажем, что она удовлетворяет ОУНБ.

Из неравенства $\frac{a - b_1}{b_2 - b_1} + \frac{b_1 \delta - \varepsilon}{b_2 - b_1} > \max\left\{\frac{\varepsilon}{b_2}, \frac{1}{b - 32^{3n}}\right\}$, во-первых вытекает, что последовательность p_2, p_3, \dots монотонно убывает, а во-вторых, что (2.36) выполняется при $i = 3$. Выполнение неравенства (2.36) тривиально при $i \geq 4$. Из леммы 2.4 следует, что построенная мартингальная мера удовлетворяет ОУНБ.

Покажем, что ОУНБ(Z) строго вложено в $\mathcal{P}(Z, \mathbf{F})$. Рассмотрим систему (2.38) совместно со следующим равенством, противоречащим ОУНБ:

$$b_3 = \frac{b_2 p_2 + \sum_{j=4}^{\infty} b_j p_j}{p_2 + \sum_{j=4}^{\infty} p_j} \iff \sum_{j=4}^{\infty} b_j p_j = b_3(1 - p_1 - p_2) - b_2 p_2.$$

Подставив это в первое уравнение системы (2.38), получим: $p_1 = \frac{b_3 - a}{b_3 - b_1}$. Второе уравнение системы (2.38) теперь имеет вид: $\sum_{j=2}^{\infty} p_j = \frac{a - b_1}{b_3 - b_1}$. Отсюда легко находим решения поставленной задачи — мартингальную меру, не удовлетворяющую ОУНБ. ■

Пример 2.1 Пусть $b_1 = 1 < a = \frac{5}{3} < b_2 = 2 < b_3 = 4 < b_4 = 8 < \dots < b_k = 2^{k-1} < \dots$. Тогда мера $P = \left(\frac{2}{3}; \frac{1}{4}; \frac{1}{16}; \frac{1}{64}; \dots\right) \in \mathcal{P}(Z, \mathbf{F})$:

$$\begin{cases} \frac{2}{3} + \sum_{i+1}^{\infty} \frac{1}{2^{i+1}} = 1 \\ \frac{2}{3} + \sum_{i=1}^{\infty} \frac{1}{2^i} = \frac{5}{3} \end{cases}$$

удовлетворяет ОУНБ.

Глава 3

Программная реализация вычисления компонент хеджирующего портфеля в случае счётного вероятностного пространства и конечнозначных цен акций

Данная глава имеет следующую структуру.

В *параграфе 3.1* представлено описание вычислительных процессов, которые необходимы для разработки программного комплекса. Показаны следующие алгоритмы:

- 1) алгоритм конструирования моделей одношаговых (\mathbf{B}, \mathbf{S}) -рынков с бесконечным числом состояний;
- 2) алгоритмы вычисления компонент мартингальных мер;
- 3) алгоритм проведения финансовых расчётов;
- 4) ряд прочих алгоритмов, необходимых для реализации программного комплекса.

В *параграфе 3.2* изложена архитектура программного комплекса.

Описаны принципы взаимодействия компонент комплекса. Рассматриваются всевозможные информационные структуры, классы, диалоги, а также дано краткое описание инструментов и библиотек, использованных при разработке.

В *параграфе 3.3* представлена работа программного комплекса. Получены и проанализированы результаты вычислений для различных моделей $(1, \mathbf{Z})$ -рынков с бесконечным числом состояний. Рассмотрены примеры вычисления компонент хеджирующего портфеля и определения справедливой цены опционов для различных случаев. А именно, рассмотрены случаи, когда:

- 1) число различных значений с.п. Z_1 равно трём и при этом любая мартингальная мера удовлетворяет ОСУХЕ.
- 2) с.п. Z_1 принимает три различных значения, но при этом мартингальные меры, удовлетворяющие ОСУХЕ, вычисляются по специальным формулам, представленным в параграфе 2.3.
- 3) с.п. Z_1 принимает любое конечное число значений, большее трёх.

3.1 Описание основных алгоритмов

Программный комплекс предназначен для вычисления компонент хеджирующих портфелей, реплицирующих финансовые обязательства марковского типа, а также определения справедливой цены опционов и выполнению других сложных расчётов, возникающих при исследовании различных моделей безарбитражных рынков со счётным числом состояний. Вычисления основаны на результатах первых двух глав диссертации.

Основной составляющей любого программного обеспечения является набор алгоритмов, возникающих на различных этапах решения постав-

ленной задачи. В процессе разработки программного комплекса был получен следующий ряд алгоритмов.

Алгоритм 1. Общий алгоритм проведения финансовых расчётов на $(1, Z)$ -рынках.

Исходные данные: вводим значения с.п. Z , определяем финансовое обязательство F_1 .

Результат: канонический хедж, реплицирующий финансовое обязательство F_1 на исходном или интерполирующем рынке.

Описание: На рисунке 5 показан укрупнённый алгоритм, согласно которому программный комплекс осуществляет вычисление компонент хеджирующего портфеля. Каждый шаг соответствует ряду вычислительных процедур.

Шаг 0. Задаём значения \mathbf{F} -адаптированного случайного процесса $Z = (Z_k, \mathbf{F}, \mathcal{F}_k)_{k=0}^1$, ($Z_0 = a$, $Z_1(B_i) = b_i$, $i \in \mathbb{N}$). Программный комплекс предполагает работу с такими моделями, когда среди значений $\{b_i\}_{i=1}^\infty$ только r ($3 \leq r < \infty$) различных: $b_1 < b_2 < \dots < b_r$, m_k — кратность значения b_k , $k = 1, 2, 3, \dots, r$, $m_k \leq \infty$. Обозначим через:

$$D_k = \bigcup_{j=1}^{m_k} B_{r(j-1)+k}, \quad k = 1, 2, \dots, r. \quad (3.1)$$

Задаём ограниченное финансовое обязательство (опцион-call, опцион-put):

$$F_1 = \sum_{k=1}^r c_k I_{D_k}.$$

Шаг 1. Проверяем финансовое обязательство на реплицируемость на исходном финансовом рынке. Если финансовое обязательство реплицируемо, то с учётом того, что в процессе проверки будет построен совершенный хедж, вычисления будут закончены (переходим на шаг 6). Иначе будем применять технику хааровских интерполяции для исходного фи-

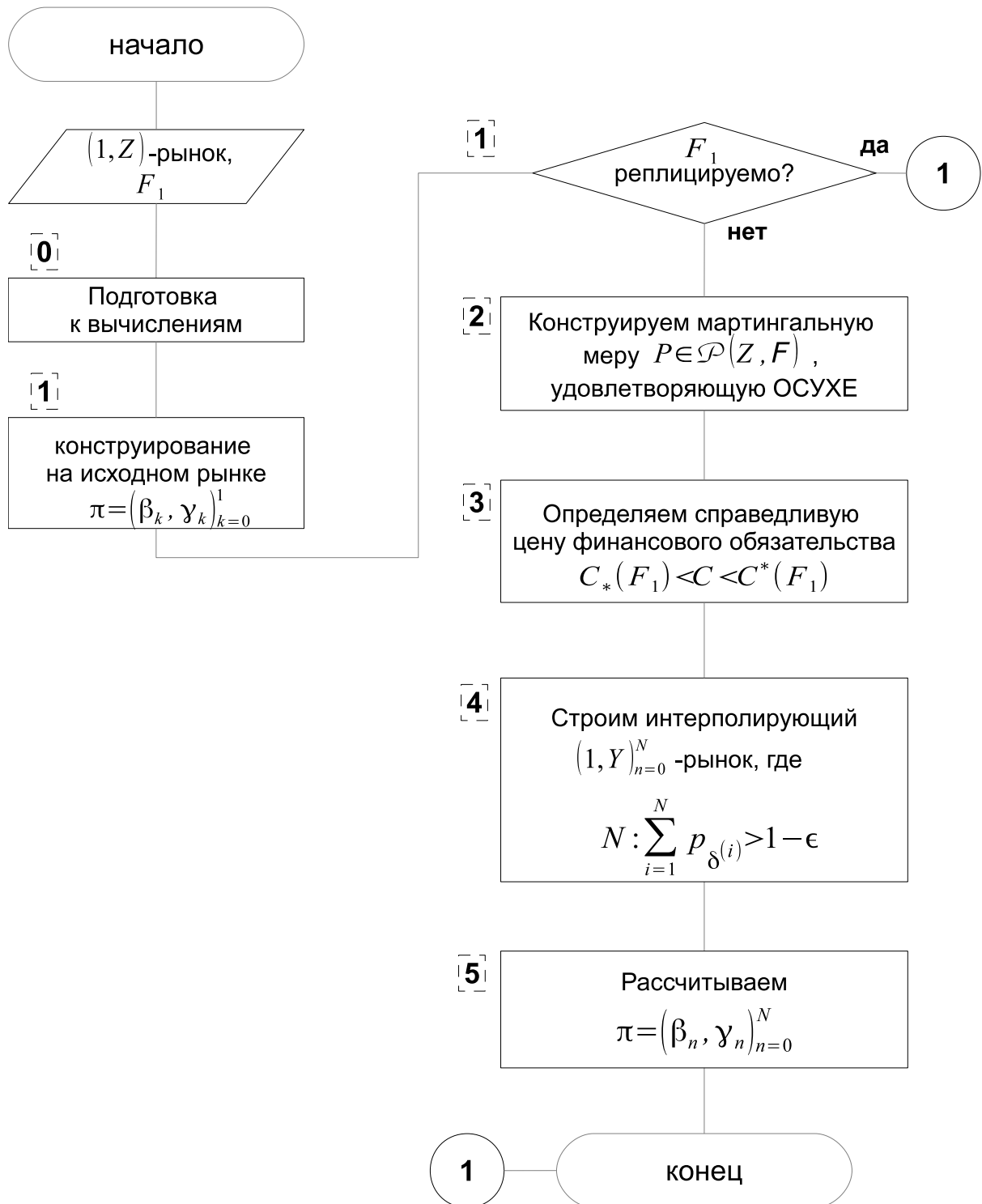


Рис. 5: Порядок ведения финансовых расчётов на $(1, Z)$ -рынках.

нансового рынка.

Шаг 2. В зависимости от модели $(1, Z)$ -рынка, построенной на шаге 0, используя результаты главы 2, найдём мартингальную меру, удовлетво-

ряющую ОСУХЕ. Так как в последовательности $\{b_i\}_{i=1}^{\infty}$ различных значений всего r , то компоненты мартингальной меры будут разбиты на r подпоследовательностей. Мы получим соответствующие значения их сумм:

$$x_k = \sum_{j=1}^{m_k} p_{r(j-1)+k}, \quad k = 1, 2, \dots, r.$$

Шаг 3. Используя, значения полученные на *шаге 2*, определим справедливую цену финансового обязательства:

$$C(F_1) = \sum_{k=1}^r c_k x_k.$$

Шаг 4. Строим специальную хааровскую интерполяцию исходного рынка. В силу того, что мы рассматриваем финансовые рынки со счётным числом состояний, мы будем использовать квантильное хеджирование. Поэтому для определения вычислительного горизонта потребуются следующие действия:

- 1) задаем точность ε ;
- 2) учитывая тот факт, что порядок появления событий B_i , $i = 1, 2, 3, \dots$ в промежуточные моменты времени не определён, поэтому представим его в виде вектора $\delta = (\delta^{(1)}, \delta^{(2)}, \delta^{(3)}, \dots)$. Координаты вектора δ — зависимые случайные величины и будут определены с помощью генератора случайных векторов, основанном на геометрическом распределении (см.[16]);
- 3) для каждого случайного события $B_{\delta^{(n)}}$, $n = 1, 2, \dots, N$ с помощью алгоритма 3 находим значения компонент мартингальной меры, полученной на шаге 2;
- 4) проверяем выполнение неравенства: $\sum_{i=1}^N p_{\delta^{(i)}} > 1 - \varepsilon$. Если неравенство верно, то N — вычислительный горизонт для интерполирующего

рынка.

По формулам, представленным в параграфе 1.3, определяем значения процесса $(Y_k)_{k=0}^N$, интерполирующего процесс Z .

Шаг 5. По формулам, представленным в параграфе 1.3, определим значения компонент хеджирующего портфеля $\pi = (\beta_n, \gamma_n)_{n=0}^N$.

Шаг 6. Конец алгоритма.

Как было сказано на шаге 4 алгоритма 1, при построении специальной хааровской интерполяции необходимо определить значение компонент $p_{\delta(n)}$ мартингальной меры, соответствующих событиям $B_{\delta(n)}$, $n = 1, 2, \dots, N$. Для этого потребуются алгоритм 2 и алгоритм 3. Структуру алгоритма 2 мы будем рассматривать на примере значения x_1 . Аналогичные рассуждения можно провести для любого значения: x_2, x_3, \dots, x_r , $3 \leq r < \infty$.

Пусть $r = 3$, тогда $x_1 = \sum_{i=1}^{m_1} p_{3i-2}$.

Алгоритм 2. Вычисление компонент мартингальной меры для интерполирующего рынка.

Шаг 0. Задаём значение x_1 , количество бросков — s .

Шаг 1. На отрезок $[0; x_1]$ бросаем точку ν , получаем отрезок $[0; \nu]$, длина отрезка $[0; \nu]$ равна значению компоненты p_1 (рис. 6).

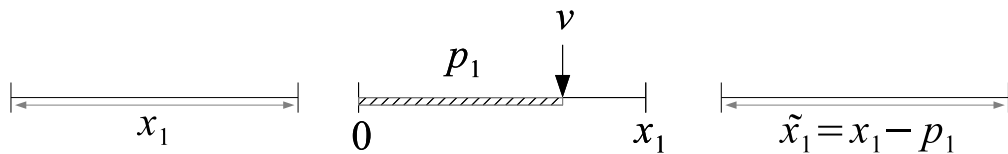


Рис. 6

Шаг 2. Бросаем на отрезок $[0; \tilde{x}_1]$ второй раз точку ν , длина отрезка $[0; \nu]$ равна значению компоненты p_4 (рис. 7).

Шаг 3. Если выполненное число бросков не превосходит заданного числа s , тогда процесс продолжается далее, иначе переходим на шаг 4.

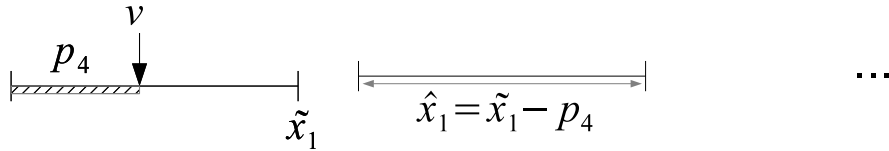


Рис. 7

Шаг 4. Конец алгоритма.

Рассмотрев структуру алгоритма 2, остаётся выяснить два вопроса:

- 1) как в программном комплексе выбирается искомое значение компоненты $p_{\delta^{(n)}}$ в промежуточные моменты времени $n = 1, 2, \dots, N$;
- 2) сколько бросков s необходимо совершить в алгоритме 2 для того, чтобы вычислить искомое $p_{\delta^{(n)}}$.

Для получения ответов на эти вопросы рассмотрим алгоритм 3. Итак, мы рассматриваем случай, когда с.п. Z_1 принимает три различных значения: b_1, b_2, b_3 с соответствующими им кратностями m_1, m_2, m_3 . Разобъём компоненту мартингальной меры $P \in \mathcal{P}(Z, \mathbf{F})$ на три подпоследовательности: $\{p_{3i-2}\}_{i=1}^{m_1}, \{p_{3i-1}\}_{i=1}^{m_2}, \{p_{3i}\}_{i=1}^{m_3}$. При этом $x_1 = \sum_{i=1}^{m_1} p_{3i-2}$; $x_2 = \sum_{i=1}^{m_2} p_{3i-1}$; $x_3 = \sum_{i=1}^{m_3} p_{3i}$. Если $m_1 = m_2 = m_3 = \infty$, тогда искомые значения компонент мартингальных мер удобно представить как это изображено на рисунке 8.

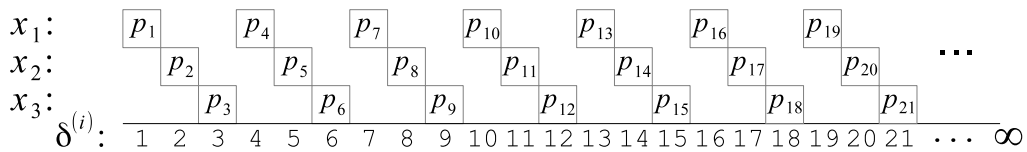


Рис. 8

Алгоритм 3.

Шаг 0. Задаём значения x_1, x_2, x_3 и соответствующие им кратности m_1, m_2, m_3 .

Шаг 1. Запускаем генератор случайных чисел, определяем значение $\delta^{(n)}$.

Шаг 2. Определим остаток от деления $\delta^{(n)}$ на 3. Полученный результат определяет значение x_k ($k = 1, 2, 3$), которому соответствует компонента $p_{\delta^{(n)}}$ (будем считать, что $k = 3$, если остаток от деления равен 0).

Шаг 3. Учитывая значения m_1, m_2, m_3 , определим существует ли компонента с полученным индексом или нет. Если компонента не существует, тогда переходим на шаг 6.

Шаг 4. Определим порядковый номер компоненты $p_{\delta^{(n)}}$ в соответствующем ряду. Обозначим его l .

Шаг 5. Определим значение компоненты $p_{\delta^{(n)}}$ по алгоритму 2, при этом количество бросков s на отрезок будем определять следующим образом:

- 1) если l меньше кратности значения $b_{\delta^{(n)}}$, тогда число бросков равно разности между значением l и количеством уже посчитанных компонент, предшествующих $p_{\delta^{(n)}}$;
- 2) если l равно кратности значения $b_{\delta^{(n)}}$, тогда количество бросков будет на единицу меньше, чем в пункте 1).

Шаг 6. Конец алгоритма.

В программном комплексе представлено 6 одношаговых моделей. Они отличаются друг от друга значениями с.п. Z . С точки зрения обозначения индексов и, следовательно, определения компонент мартингальных мер, выделяется случай 1. На рисунке 9 показана схема компонент мартингальной меры для этого случая. Заметим, что $p_1 = x_1$ и для определения этой компоненты мы не будем использовать алгоритм 2. Для определения компонент соответствующих x_2 и x_3 ($m_1 = m_2 = \infty$) необходимо изменить шаг 3 алгоритма 3 следующим образом:

Шаг 3. Определим остаток от деления $\delta^{(1)}$ на 2. Полученный результат

определяет номер $k = 2, 3$ (будем считать, что $k = 3$, если остаток от деления равен 1, $k = 2$, если остаток от деления равен 0).

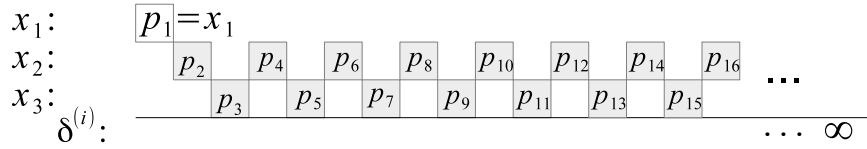


Рис. 9

Для полной ясности рассмотрим 3 итерации алгоритма 3 на следующем примере.

Пример 3.1 *С.п. Z_1 принимает три различных значения: b_1, b_2, b_3 ; с соответствующими им значениями кратностей: $m_1 = 5, m_2 = m_3 = \infty$.*

Схема компонент мартингальной меры показана на рисунке 10.

Итерация 1.

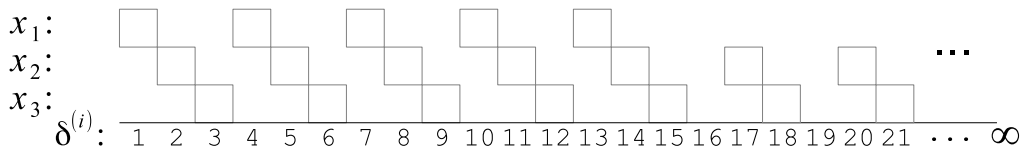


Рис. 10

Шаг 1. Пусть $\delta^{(1)} = 11$ (т.е. необходимо вычислить p_{11}).

Шаг 2. Компонента p_{11} соответствует x_2 .

Шаг 3. С учётом того, что $m_2 = \infty$, делаем вывод, что компонента $p_{\delta^{(11)}}$ существует.

Шаг 4. $l = 4$ (рис. 11).

Шаг 5. Так как $11 < \infty$ и число уже посчитанных компонент, предшествующих p_{11} , равно 0, то число бросков $s = 4$. Применяем алгоритм

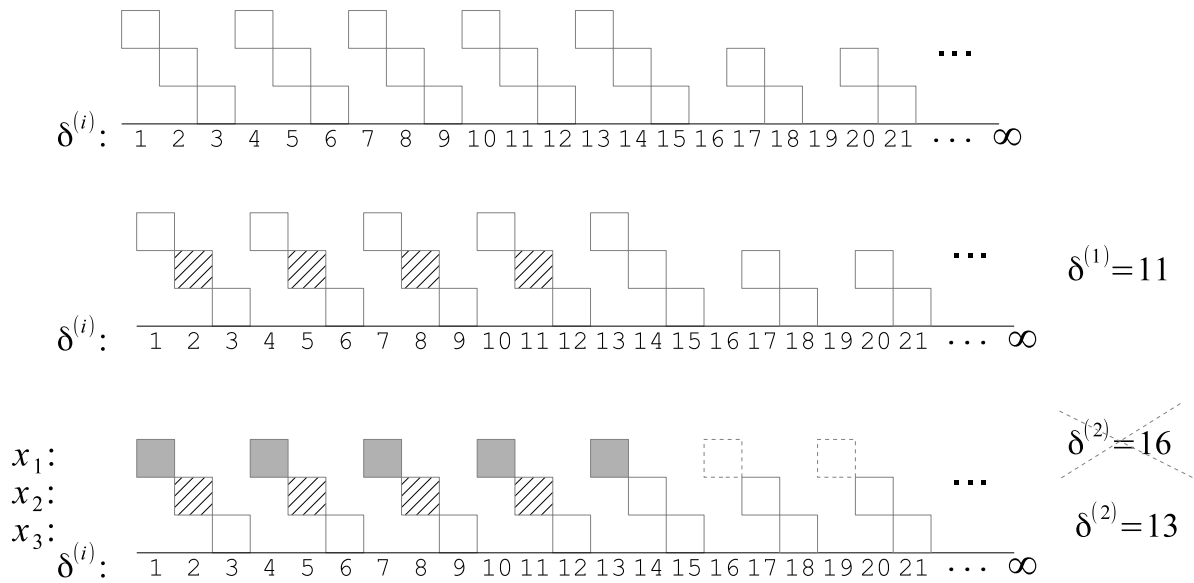


Рис. 11

2, находим значение p_{11} .

Шаг 6. Конец алгоритма.

Итерация 2.

Шаг 1. Пусть $\delta^{(2)} = 16$ (т.е. необходимо вычислить p_{16}).

Шаг 2. Компонента p_{16} соответствует x_1 .

Шаг 3. С учётом того, что $m_1 = 5$, делаем вывод, что компонента p_{16} не существует (рис. 11).

Шаг 6. Конец алгоритма.

Итерация 3.

Шаг 1. Пусть $\delta^{(2)} = 13$ (т.е. необходимо вычислить p_{13}).

Шаг 2. Компонента p_{13} соответствует x_1 .

Шаг 3. Компонента p_{13} существует.

Шаг 4. $l = 5$ (рис. 11).

Шаг 5. Так как $l = m_1$ и число уже посчитанных компонент, предшествующих p_{13} , равно 0, то число бросков $s = 4$. Применяем алгоритм 2, находим значение p_{13} (значение p_{13} равно длине последнего отрезка в

алгоритме 2, т.е. $p_{13} = x_1 - \sum_{i=1}^4 p_{3i-2}$

Шаг 6. Конец алгоритма.

Алгоритм 4. Алгоритм построения мартингальных мер, удовлетворяющих ОСУХЕ, когда с.п. Z принимает три значения: $b_1 < a < b_2 < b_3$, $m_1 = 1, m_2 = m_3 = \infty$.

Шаг 0. Вводим значение с.п. Z : $b_1 < a < b_2 < b_3$, задаём финансовое обязательство $F_1 = \sum_{i=1}^3 c_i I_{D_i}$ и цену исполнения K .

Шаг 1. Определим решение задачи:

$$C = \sum_{i=1}^3 c_i x_i \longrightarrow \min$$

$$\begin{cases} \sum_{i=1}^3 x_i = 1 \\ \sum_{i=1}^3 b_i x_i = a \\ x_i > 0, i = 1, 2, 3 \end{cases}$$

Шаг 2. Используя полученное на шаге 1 решение, определим

$$C_* = \sum_{i=1}^3 c_i x_i.$$

Шаг 3. Определим решение задачи:

$$C = \sum_{i=1}^3 c_i x_i \longrightarrow \max$$

$$\begin{cases} \sum_{i=1}^3 x_i = 1 \\ \sum_{i=1}^3 b_i x_i = a \\ x_i > 0, i = 1, 2, 3 \end{cases}$$

Шаг 4. Используя полученное на шаге 3 решение, определим

$$C^* = \sum_{i=1}^3 c_i x_i.$$

Шаг 5. Определим решение задачи:

$$\begin{cases} \sum_{i=1}^3 x_i = 1 \\ \sum_{i=1}^3 b_i x_i = a \\ \sum_{i=1}^3 c_i x_i = \frac{C_* + C^*}{2} \\ x_i > 0, \quad i = 1, 2, 3 \end{cases}$$

Шаг 6. Конец алгоритма.

Согласно теореме 2.1 мартингальная мера, полученная на шаге 5, удовлетворяет ОСУХЕ.

Алгоритм 5. Алгоритм построения мартингальных мер, удовлетворяющих ОСУХЕ, когда с.п. Z_1 принимает r ($3 < r < \infty$) рациональных значений.

Рассмотрим систему:

$$\begin{cases} \sum_{i=1}^r x_i = 1 \\ \sum_{i=1}^r b_i x_i = a \\ x_i > 0, \quad i = 1, 2, 3, \dots, r \end{cases} \quad (3.2)$$

Шаг 0. Вводим значение r ; значения с.п. Z : $a, b_1, b_2, b_3, \dots, b_r$; $m_1, m_2, m_3, \dots, m_r$; задаём финансовое обязательство $F_1 = \sum_{i=1}^r c_i I_{D_i}$ и цену исполнения K .

Шаг 1. Определим значение числа l : $b_l < a < b_{l+1}$ ($1 \leq l \leq r - 1$). В случае, если $m_l = m_{l+1} = \infty$, тогда переходим на шаг 2, иначе (одно из значений m_l, m_{l+1} — конечное), переходим на шаг 3.

Шаг 2. Имеем: $m_l = m_{l+1} = \infty, m_k \leq \infty, 1 \leq k \leq r, k \neq l, l + 1$.

Вычислим $n = \max_{1 \leq i \leq r} \left[\frac{\sum_{\substack{k=1, \\ k \neq l, l+1}}^r |b_i - b_k|}{|b_i - a|} \right] + 1$.

Вычислим $x_k = \frac{1}{n}$, $1 \leq k \leq r$, $k \neq l, l+1$.

$$\tilde{1} = 1 - \frac{r-2}{n};$$

$$\tilde{a} = a - \frac{1}{n} \sum_{\substack{k=1, \\ k \neq l, l+1}}^r b_k;$$

$$\tilde{b}_l = \tilde{1} \cdot b_l$$

$$\tilde{b}_{l+1} = \tilde{1} \cdot b_{l+1}.$$

Решая систему (3.2), получим:

$$x_l = \frac{\tilde{b}_{l+1} - \tilde{a}}{\tilde{b}_{l+1} - \tilde{b}_l}; \quad x_{l+1} = \frac{\tilde{a} - \tilde{b}_l}{\tilde{b}_{l+1} - \tilde{b}_l}.$$

Шаг 3. Имеем: одно из значений m_l, m_{l+1} — конечное, $m_k \leq \infty$, $1 \leq k \leq r$, $k \neq l, l+1$ (как минимум одно из них бесконечное).

$$\text{Вычислим } t = \max_{1 \leq i \leq r} \left[\frac{\sum_{\substack{k=1, \\ k \neq l, l+1}}^r |b_i - b_k|}{|b_i - a|} \right]; n = t^2 + \frac{1}{2}.$$

Вычислим $x_k = \frac{1}{n}$, $1 \leq k \leq r$, $k \neq l, l+1, m$, где $1 \leq m \leq r$ — произвольный индекс; $x_m = \frac{1}{\sqrt{n}}$.

$$\tilde{1} = 1 - \frac{r-3}{n} - \frac{1}{\sqrt{n}};$$

$$\tilde{a} = a - \frac{1}{n} \sum_{\substack{k=1, \\ k \neq l, l+1, m}}^r b_k - \frac{1}{\sqrt{n}} b_m;$$

$$\tilde{b}_l = \tilde{1} \cdot b_l;$$

$$\tilde{b}_{l+1} = \tilde{1} \cdot b_{l+1}.$$

Решая систему (3.2), получим:

$$x_l = \frac{\tilde{b}_{l+1} - \tilde{a}}{\tilde{b}_{l+1} - \tilde{b}_l}; \quad x_{l+1} = \frac{\tilde{a} - \tilde{b}_l}{\tilde{b}_{l+1} - \tilde{b}_l}.$$

Шаг 4. Конец алгоритма.

Используя результаты теоремы 2.3, можно доказать, что построенная мартингальная мера удовлетворяет ОУНБ.

3.2 Архитектура программного комплекса

Основные положения

Комплекс состоит из основной программы (загрузчика модулей) и подгружаемых модулей (плагинов¹), представленных в виде динамически подключаемых библиотек (Dynamic Link Libraries). Плагины реализуют графические интерфейсы и расчетную часть для конкретных моделей. Такая логическая организация позволяет развивать и дополнять программный комплекс новыми моделями и вычислительными процедурами. При работе над настоящей диссертацией был создан модуль — "Одношаговые модели (B,S)-рынков со счетным числом состояний" (рис. 12). Некоторые алгоритмы, лежащие в его основе мы и рассмотрели в параграфе 3.1.

Исходный код комплекса организован по-принципу мини-фреймворка² (далее просто фреймворк) таким образом, что предоставляет пишущему модуль программисту базовый набор примитивов, необходимых для организации работы с деревьями событий случайных процессов, таких как (1, Z)-рынки. А именно, произведении расчетов и визуализации этих деревьев в виде схем. Помимо этого, фреймворк содержит базовые элементы управления, позволяющие программисту создавать из них (как из кубиков) пользовательский интерфейс с учетом нюансов вычислений, проводимых в рамках конкретной модели. На рисунке 13 схематически показана концептуальная модель программного комплекса.

При разработке программного комплекса использовались принципы

¹Плагин (plug-in, «подключать») — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей

²фреймворк (framework) — программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

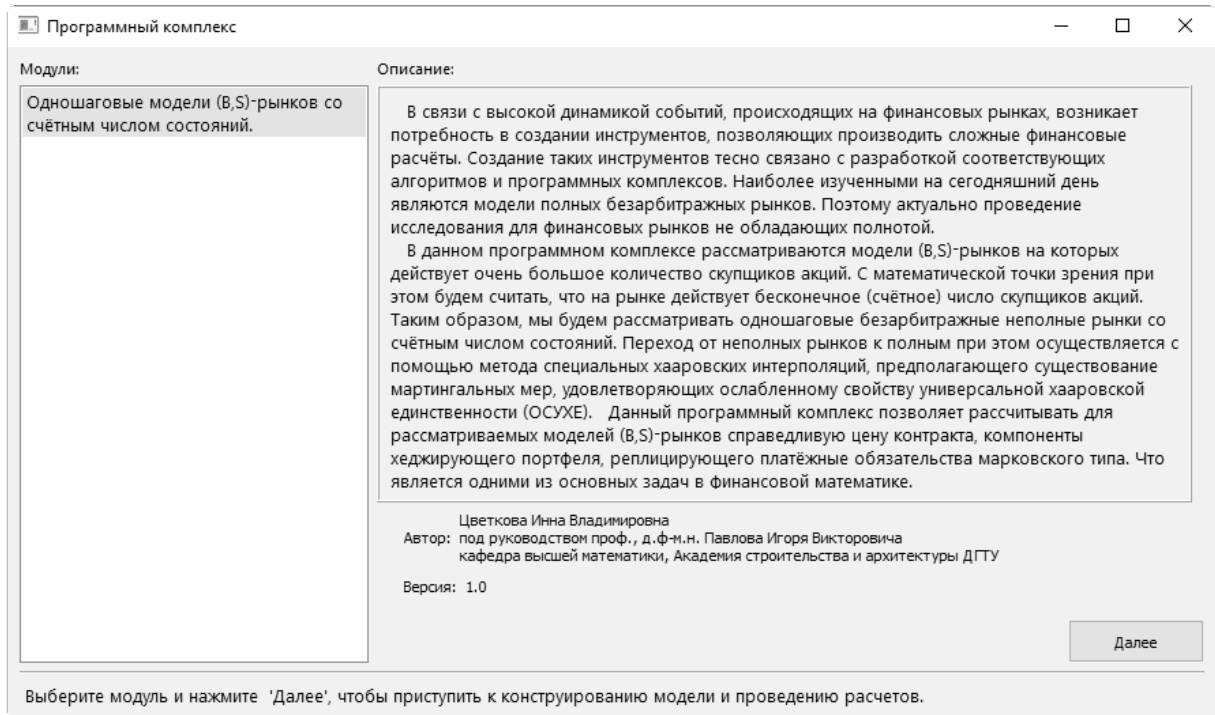


Рис. 12: Модуль — "Одношаговые модели (B,S)-рынков со счетным числом состояний".

объектно-ориентированного программирования. На этапе проектирования был выделен основной функционал и функционал, присущий реализации конкретной модели. В результате был получен набор классов, UML³-диаграммы отношений которых представлены на рисунках 15, 16 и 17. Приведем краткое описание семантики используемых на диаграммах отношений между классовыми блоками.

Ассоциация — объекты одного класса связаны с объектами другого таким образом, что можно перемещаться от объектов одного класса к другому. *На нашей диаграмме классов представлены двойные ассоциации, обозначенные сплошными линиями без стрелок между блоками классов.*

Агрегация — один класс является контейнером для других классов,

³UML (Unified Modeling Language) — это унифицированный графический язык моделирования для описания, визуализации, проектирования и документирования объектно-ориентированных систем.

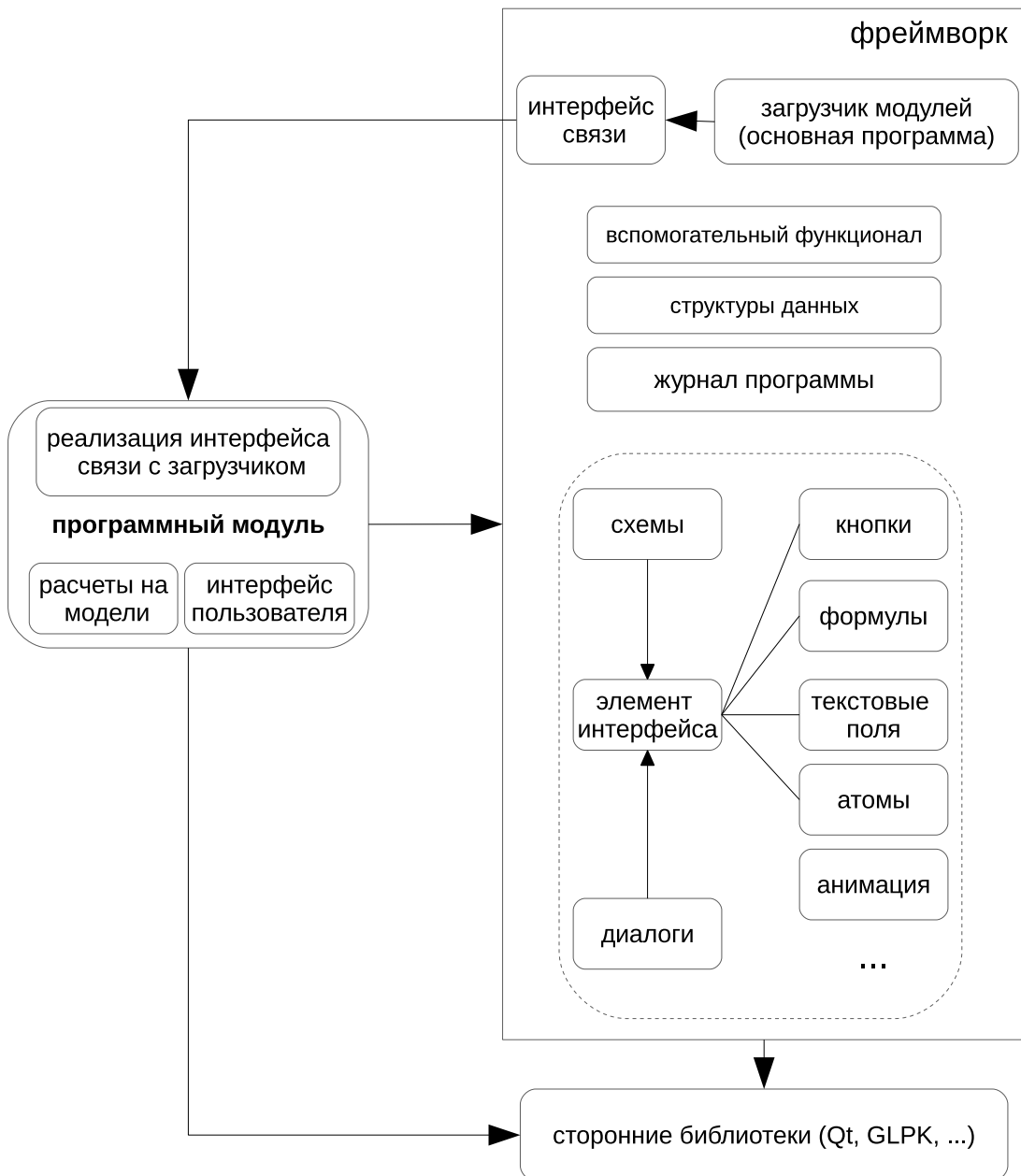


Рис. 13: Программный комплекс как фреймворк для программирования.

причем время существования содержащихся классов не зависит от времени существования содержащего их контейнера. На диаграмме агрегация представлена пустым ромбом на блоке класса-контейнера, и линией, идущей от этого ромба к содержащемуся классу.

Композиция — более строгий вариант агрегации, когда жизненный цикл экземпляров содержащихся классов зависит от жизни класса-контейнера. То есть, при уничтожении контейнера всё его содержимое

будет также уничтожено. *На диаграмме выглядит как агрегация, но имеет закрашенный ромб.*

Обобщение — это наследование одного класса другим. *На диаграмме обобщение представлено линией с пустым треугольником у суперкласса (наследуемого класса).*

Реализация — отношение между двумя классами при котором один класс (абстрактный, класс-интерфейс) задает требования к поведению системы (функциональную спецификацию), а другой является реализацией этого поведения. *На диаграмме реализация представляется так же, как и наследование, но с пунктирной линией.*

Зависимость — это слабая форма отношения использования («use»), при котором изменение в спецификации одного класса (независимого) влечёт за собой изменения в другом (зависимом). Например, такая форма отношения возникает тогда, когда объект выступает в форме параметра или локальной переменной. *Графически представляется штриховой стрелкой, идущей от зависимого элемента к тому, от которого он зависит.*

Классы, имена которых начинаются с RC (обозначение введенное нами как сокращение от Research Complex) являются частью разработанного фреймворка, их блоки на диаграмме имеют серый фон. Прочие классы являются частью программного модуля (далее просто модуль или плагин) и имеют белый фон. *Чтобы исключить избыточность, на приведенных диаграммах не указаны взаимоотношения между разработанными нами классами и классами фреймворка Qt.*

RCTree — класс-контейнер, содержащий данные и методы для работы с деревьями событий $(1, Z)$ -рынков. Он содержит множество экземпляров класса *RCTreeFiltration* (столько, сколько шагов у модели) и корневой узел *RCTreeNode*.

RCTreeFiltration разработан для того, чтобы была возможность однозначно определить число и порядок атомов на каждом шаге модели. Этот класс-контейнер содержит упорядоченный (сверху вниз) список узлов (*RCTreeNode*) и методы доступа к ним по индексу.

Класс *RCTreeNode* предназначен для организации работы с атомом (событием) информационного дерева $(1, Z)$ -рынка. Класс спроектирован таким образом, что позволяет содержать исходные данные (заданные на атоме), различные результаты расчетов и информацию о связи с узлом-родителем и всех его узлах-потомках.

Для пояснения, приведем схему одношаговой модели (рис. 14) из примеров, которые будут даны далее в параграфе 3.3. В программе атомы Ω , D_1 , D_2 , D_3 будут реализованы как экземпляры класса *RCTreeNode*. А экземпляр *RCTree* будет содержать два класса-контейнера *RCTreeFiltration* (для нулевого и первого шага).

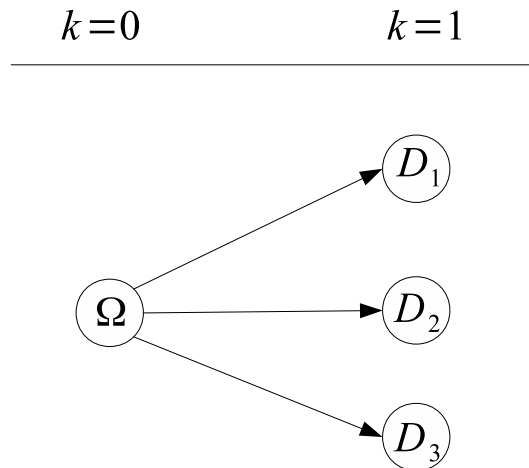


Рис. 14

Данные классы предназначены только для проведения расчетов и хранения результатов. Что позволяет создавать столько экземпляров этих структур, сколько нужно для решения поставленной задачи. Для отображения же схемы дерева в пользовательском интерфейсе разработаны

следующие классы.

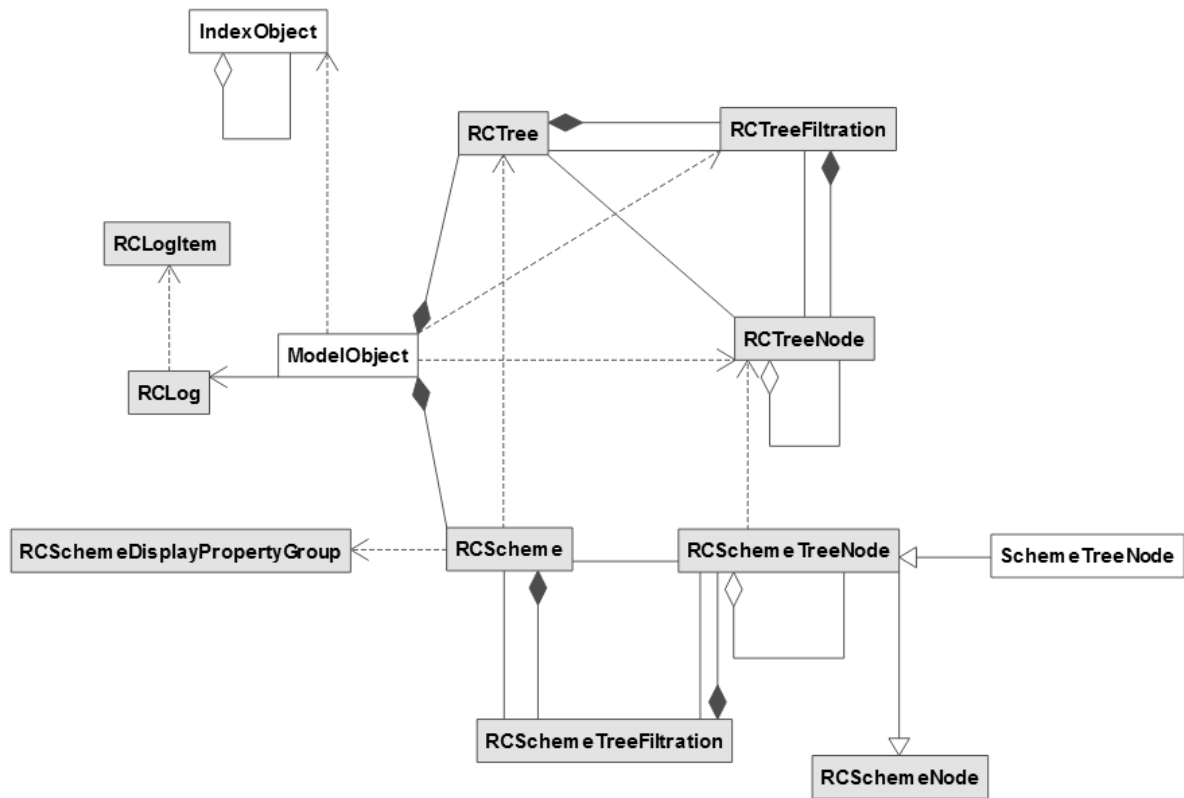


Рис. 15: Взаимосвязь основных классов программного комплекса (UML-схема 1).

RCScheme — определяет структуру схемы, которая должна быть показана пользователю. *RCSchemeTreeFiltration* — содержит упорядоченный (сверху вниз) список узлов *RCSchemeTreeNode* и методы доступа к ним по индексу. *RCSchemeTreeNode* — определяет правила для отображения атома. Отметим, что объекту *RCScheme* может быть назначен экземпляр класса *RCTree*, однако это не обязательно. Так, примеры отображения схем не привязанных к структуре *RCTree* можно увидеть в параграфе 3.3, в диалогах выбора случая и задания параметров модели. Если же такая привязка была осуществлена, тогда *RCScheme* будет отображать актуальные данные привязанного к нему *RCTree*. Экземпляры класса-посредника *RCSchemeDisplayPropertyGroup*, назначенные объекту *RCScheme* позволяют контролировать какие данные *RCTreeNode*

должны быть показаны пользователю на отображаемой схеме дерева. Например, пользователь может пожелать узнать значение компонент посчитанной вероятностной меры или дисконтированной цены актива и т.п.

Для вывода подробностей проводимых расчетов в программе предусмотрен журнал сообщений.

RCLog — класс, реализующий работу с журналом сообщений и предоставляющий программисту модуля методы для добавления сообщений в журнал. Внутри реализации данного класса сообщение представлено в виде структуры (*RCLogItem*), состоящей из тела сообщения и времени его записи в журнал.

RCLogWidget — элемент графического интерфейса, показывающий все записи внесенные в журнал. В программном комплексе отображается в отдельном окне.

ModelObject — реализует все вычислительные процедуры, представленные в параграфе 3.3, конструирует *RCTree* и конструирует для них "Одношаговые модели (B,S)-рынков со счетным числом состояний".

IndexObject — класс реализует алгоритм вычисления значений компонент меры $p_{\delta(i)}$, описанный в параграфе 3.1. Для каждой из m_i ($i = 1, 2, \dots, r$) создается свой экземпляр класса, отвечающий за вычисление компонент меры, согласно введенным нами обозначениям. Управление же этими экземплярами осуществляется через статические методы класса-контейнера, созданного по-типу паттерна Одиночка (англ. Singleton). Таким образом, класс *IndexObject* является контейнером для экземпляров самого себя. Что это дает? Рассмотрим небольшой пример C++ кода, использованного при вычисления интерполяционного горизонта и компонент меры в случае 3 (подробнее про различные случаи смотрите параграф 3.1 и параграф 3.3, полный код класса приведен в приложении к настоящей диссертации).

```

IndexObject::Initialize(3); // у нас три различных m
// объект для m1, -1 означает, что m1 = бесконечности
IndexObject::AddObject(new IndexObject(0, -1, x1));
IndexObject::AddObject(new IndexObject(1, m2_, x2)); // объект для m2
// объект для m3, -1 означает, что m3 = бесконечности
IndexObject::AddObject(new IndexObject(2, -1, x3));

int N = 1;

(начало цикла)
... (цикл, пока для текущей суммы p_i выполняется: p_sum < 1 - epsilon)
    double p_out; // сюда будет помещена очередная
                  // вычисленная компонента

    // получаем значение с.в. (генератор на основе
    // геометрического распределения)
    int delta_i = GeometricDistribution();

    if (!IndexObject::AddNewIndex(delta_i, &p_out)) {
        // такой индекс не существует или уже выпадал
        // нужно новое значение с.в. delta_i
        continue;
    }

    p_sum += p_out;
    N = N + 1; // переходим на следующий шаг интерполяции
    ...
(конец цикла)

// на выходе из цикла получено значение горизонта N

// здесь и далее легко по индексу получит значение посчитанной
// компоненты меры для дальнейших вычислений
IndexObject::GetMeasure(i, &p_exists);

```

Для отрисовки пользовательского интерфейса используются технология `QGraphicsScene / QGraphicsView` из фреймворка Qt. В частности, класс `RCGraphicsScene` наследован от `QGraphicsScene`, а `RCGraphicsView` от `QGraphicsView` соответственно. Класс `QGraphicsScene` предоставляет

турой *RScheme*.

RCAtomGraphicsItem — рисует узел дерева (поддерживает математические формулы, см. класс *RCMathGraphicsItem*), позволяет пользователю редактировать данные на узле.

RCArrowGraphicsItem — рисует на схеме дерева направляющие стрелки от узлов-родителей к узлам-потомкам.

RCMathGraphicsItem — рисует математическую формулу, используя вспомогательные классы-мосты (*RCMathMLBridge* и *RCMathMLBridgePrivate*) для доступа к функциям внедренного в комплекс парсера языка MathML (*QwtMathMLDocument*). Классы-мосты позволяют программному комплексу абстрагироваться от конкретной реализации парсера, который в дальнейшем может быть заменен другим.

RCMathMLTemplates — шаблоны для языка MathML, используемые в программе.

RCSchemeTimelineItem — отображение шагов модели для конкретной схемы дерева.

RCGraphicsButton — класс-кнопка (все кнопки, представленные на снимках экрана в параграфе 3.3 - экземпляры этого класса).

RCGraphicsTextArea — текстовая информация на полупрозрачном фоне, всплывающая поверх всего содержимого окна. Используется для важных сообщений при проведении расчетов, а также для сообщения подробностей при щелчке мышкой на сносках (например, детальное описание литературного источника).

RCBigArrowGraphicsItem — крупная стрелка.

RCAnimation — используется другими элементами управления, для плавной смены позиции или появления на экране (пропадания с экрана).

RCCalculationDialog — базовый диалог для организации вычислений, со-

держит все необходимые элементы управления (список шагов основного алгоритма, детальное пояснение конкретного шага, кнопки для отображения различных данных и т.д.).

CalculationDialog — наследует *RCCalculationDialog*, уточняя его реализацию для вычислений на конкретной модели (см. снимки экрана этого диалога в параграфе 3.3). Реализован в плагине.

StartDialog — это диалоговое окно создается после запуска плагина и предназначено для выбора модели, случая и ввода исходных данных (см. снимки экрана этого диалога в параграфе 3.3). Реализован только в плагине, не имеет базового класса в разработанном фреймворке.

Schemes — содержит все схемы информационных деревьев (объекты класса *RCScheme*), использованные в *StartDialog*.

Для подключения к основной программе-загрузчику все плагины должны реализовывать интерфейс *RCPluginInterface*, который является абстрактным классом (см. рисунок 17). Подключение осуществляется в классе *RCApplication*.

PluginObject — наследует и реализует класс *RCPluginInterface*, содержит сведения о разработчике модуля.

Инструменты разработчика

Основные языки программирования, использованные при разработке — это С и С++. Разработка велась в среде Qt Creator, которая является надстройкой над фреймворком Qt5. Дополнительно использовались:

- открытая (Open Source) библиотека функции для решения задач линейного программирования (GLPK – GNU Linear Programming Kit);

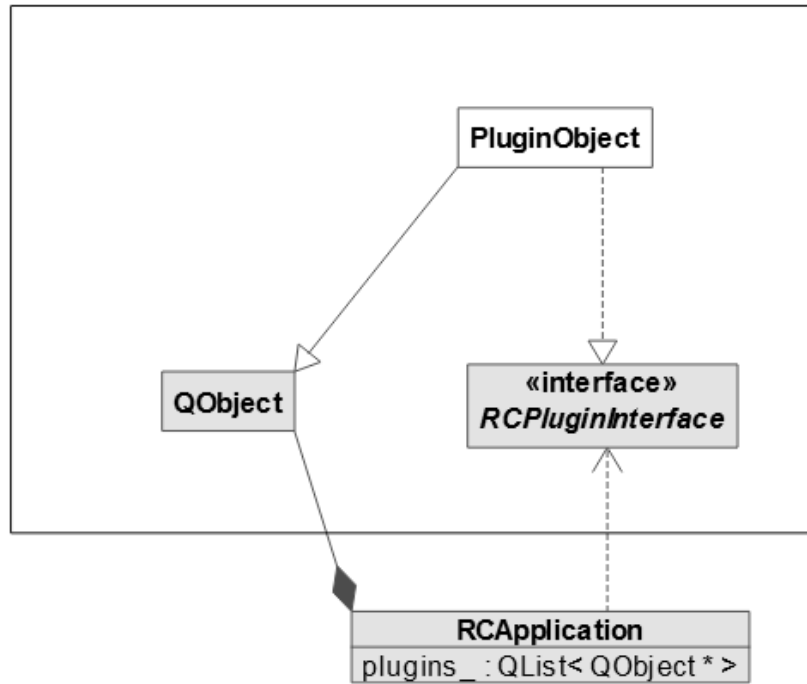


Рис. 17: Подключения плагина к основной программе (UML-схема 3).

- парсер⁴ и визуализатор языка MathML⁵ из открытой библиотеки Qwt (основана на C++ / Qt), который был модифицирован таким образом, что стал пригоден для работы в составе программного комплекса (в результате этого в нем были исправлены и некоторые ошибки отрисовки). Все формулы, представленные в программе, в исходном коде записаны на языке MathML.

Это позволило обеспечить выполнение следующих требований к итоговой программе:

1. высокая скорость исполнения программного кода вычислительной части программы (языки C и C++);
2. реализация простого и, во многом, интуитивно понятного пользова-

⁴парсер (parser, синтаксический анализатор) — часть программы, преобразующей входные данные (как правило, текст) в структурированный формат.

⁵MathML (Mathematical Markup Language — «язык математической разметки») — это язык разметки для представления математических символов и формул в документах сети интернет.

- тельского интерфейса (диалоги, схемы, формулы, подсказки и т.д.);
3. возможность переноса программы на другую платформу с минимальными усилиями;
 4. возможность расширения программного комплекса для создания новых моделей и организации вычислений на них, путем повторного использования разработанных классов и объектов, объединенных в мини-фреймворк.

Для вычислений использовались типы данных `int` (`integer` – целое, для операций с целыми числами) и `double` (число двойной точности, для операций с дробными числами). Отметим, что числа двойной точности с плавающей запятой обеспечивают точность в 15–17 десятичных цифр.

Программный комплекс поддерживает все системы семейства Windows, начиная с Windows XP (т.е. Windows XP/Vista/7/8/8.1/10).

3.3 Результаты расчетов на примерах

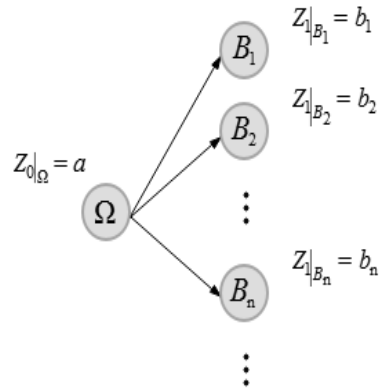
В программном комплексе рассмотрено 6 различных моделей одношаговых (\mathbf{B}, \mathbf{S}) -рынков со счётным числом состояний (рис. 18). В примерах (3.2), (3.3) будут представлены результаты вычислений компонент хеджирующего портфеля а также пользовательский интерфейс программного комплекса.

Пример 3.2 *Значения с.п. Z : $b_1 < a < b_2 < b_3$; $m_1 = 1$, $m_2 = m_3 = \infty$ (рис. 19).*

Согласно обозначению (3.1), рассмотрим события D_1, D_2, D_3 (рис. 20): Введём значения с.п. Z , выбираем тип финансового обязательства и цену исполнения контракта (рис. 21). Проверяем финансовое обязательство

- Одношаговая модель
- Случай 1
- Случай 2
- Случай 3
- Случай 4
- Случай 5
- Случай 6

Дерево стохастического базиса одношагового (B,S)-рынка



Ω – счётное пространство элементарных событий, которое при переходе от момента времени $k=0$ к $k=1$ разбивается на счётное число атомов $B_i, i=1,2,3,\dots$.

Z – случайный процесс, который мы мыслим как дисконтированную стоимость акции.

В зависимости от значений случайной величины Z_1 можно выделить следующие случаи.

Рис. 18

на реплицируемость на исходном рынке (рис. 22). В результате получаем справедливую цену контракта, значения компонент канонического хеджа, его полный капитал (рис. 23, 24, 25, 26). Аналогичный интерфейс имеют все остальные 5 случаев.

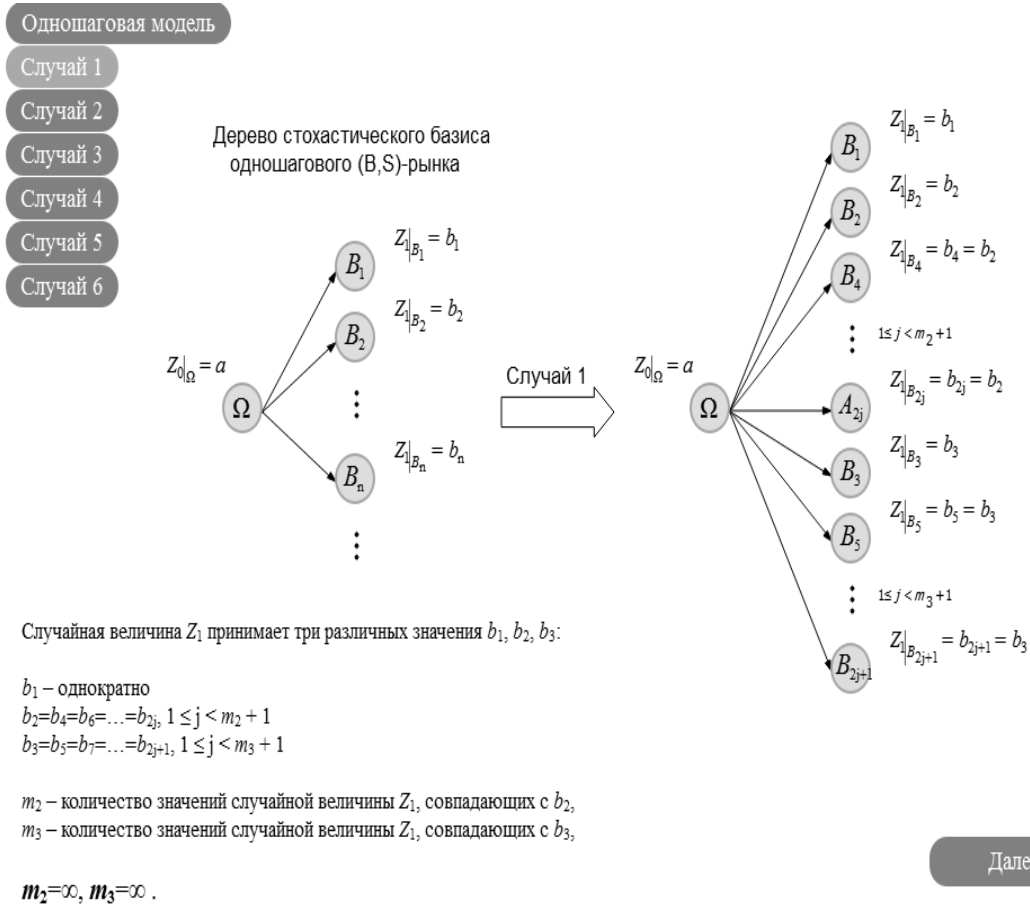
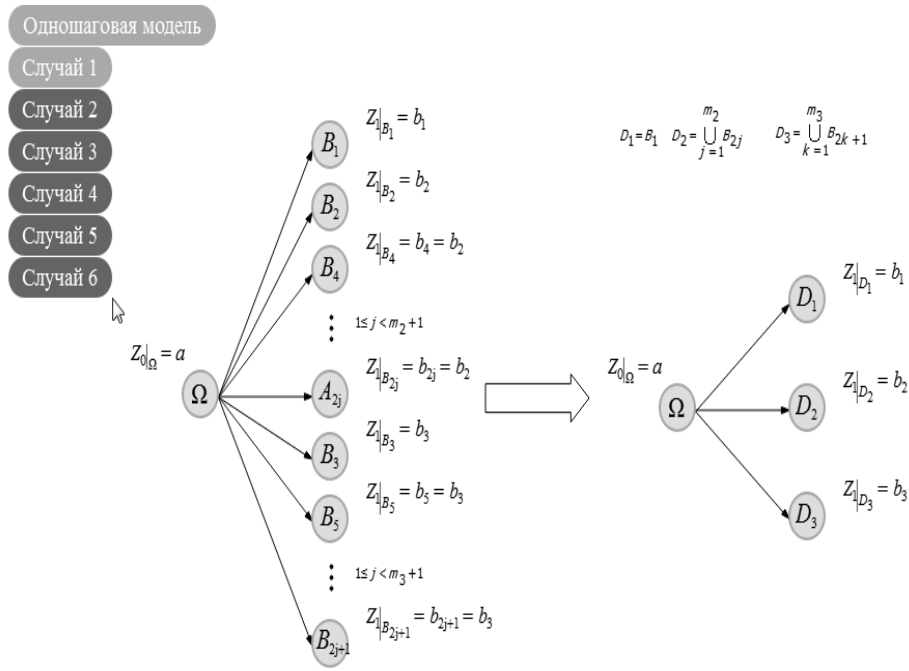


Рис. 19

Пример 3.3 Рассмотрим результаты вычислений канонического хеджа в случае 2. Значения с.п. Z : $a = 5$; $b_1 = 3$; $b_2 = 7$; $b_3 = 10$; $m_1 = m_2 = m_3 = \infty$ (рис. 27,28).

В результате проверки заданного финансового обязательства на исходном рынке, получаем, что оно нереплицируемо. Поэтому переходим к вычислению мартингальной меры, удовлетворяющей ОСУХЕ (используем формулы из параграфа 2.3). Вводим значение параметра $n = 10$ (рис. 29). Получаем значения мартингальной меры: x_1, x_2, x_3 и справедливую цену контракта (рис. 30). Далее следуя алгоритму 1 вводим значение ε (рис. 31). Согласно алгоритму 2, 3 вычисляем компоненты мартингальной меры (рис. 32).



Так как производить расчёты на модели с бесконечным числом атомов технически неудобно, поэтому выделим три подгруппы событий, на каждом из которых значения случайной величины Z_1 совпадают с b_1, b_2, b_3 соответственно. Получим D_1, D_2, D_3 .

Назад

Далее

Рис. 20

В результате получаем значения компонент канонического хеджа, его полный капитал (рис. 33, 34, 35).

Одношаговая модель

- Случай 1
- Случай 2
- Случай 3
- Случай 4
- Случай 5
- Случай 6

$$D_1 = b_1 \quad D_2 = \bigcup_{j=1}^{m_2} b_{2j} \quad D_3 = \bigcup_{k=1}^{m_3} b_{2k+1}$$

Параметры

$m_2 =$ $m_3 =$

$a =$ $b_1 =$ $b_2 =$ $b_3 =$

$b_1 < a < b_2 < b_3$

Задать платежное обязательство:

цена исполнения =

Введите значения случайного процесса Z , задайте тип платёжного обязательства и нажмите "Начать вычисления".

Назад

Начать
вычисления

Рис. 21

Шаг 1. Проверка финансового обязательства на реплицируемость.

Шаг 2. Вычисление цены финансового обязательства и компонент мартингальной меры.

Шаг 3. Построение специальной хааровской интерполяции.

Шаг 4. Вычисление компонент самофинансируемого портфеля.

Шаг 5. Конец.

Исходный процесс

$$\begin{cases} \beta_1 + \gamma_1 Z_1(D_1) = f(D_1) \\ \beta_1 + \gamma_1 Z_1(D_2) = f(D_2) \\ \beta_1 + \gamma_1 Z_1(D_3) = f(D_3) \\ \beta_1 + \gamma_1 Z_0(\Omega) = X_0 \end{cases}$$

Проверяя систему на реплицируемость на исходном рынке решая

Если рассматриваемая система имеет решение мы получаем портфель λ и на этом шаге вычисления завершаются (конец алгоритма). Если же финансовое обязательство нереплицируемо, то рассматриваемый рынок является неполным. Переход полному рынку будем осуществлять с помощью метода специальных хааровских интерполяций.

Алгоритм

Результаты

Описание

Следующий шаг

Рис. 22

Финансовое обязательство f реплицируемо на исходном рынке.

Рис. 23

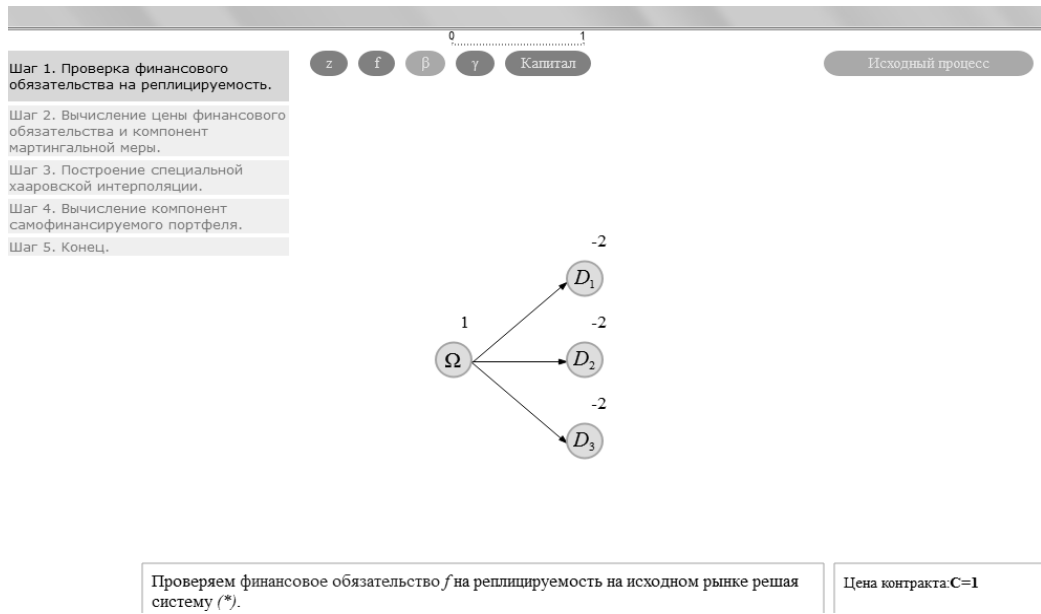


Рис. 24

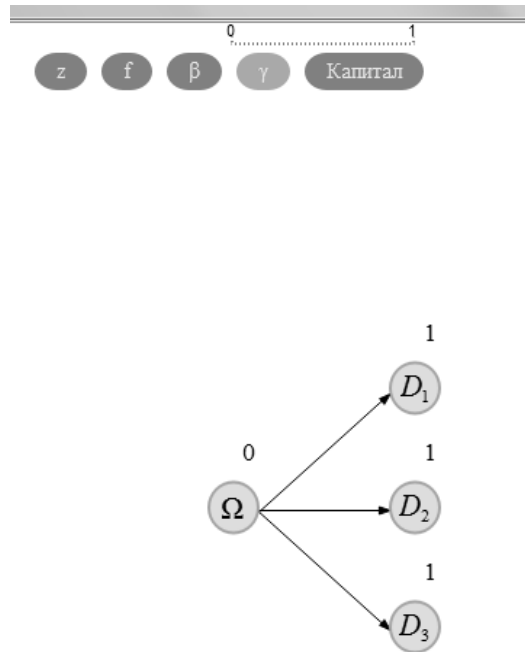


Рис. 25

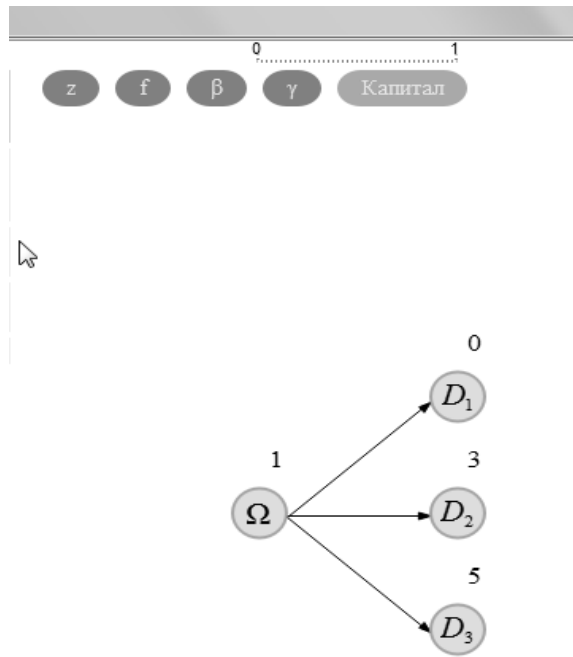


Рис. 26

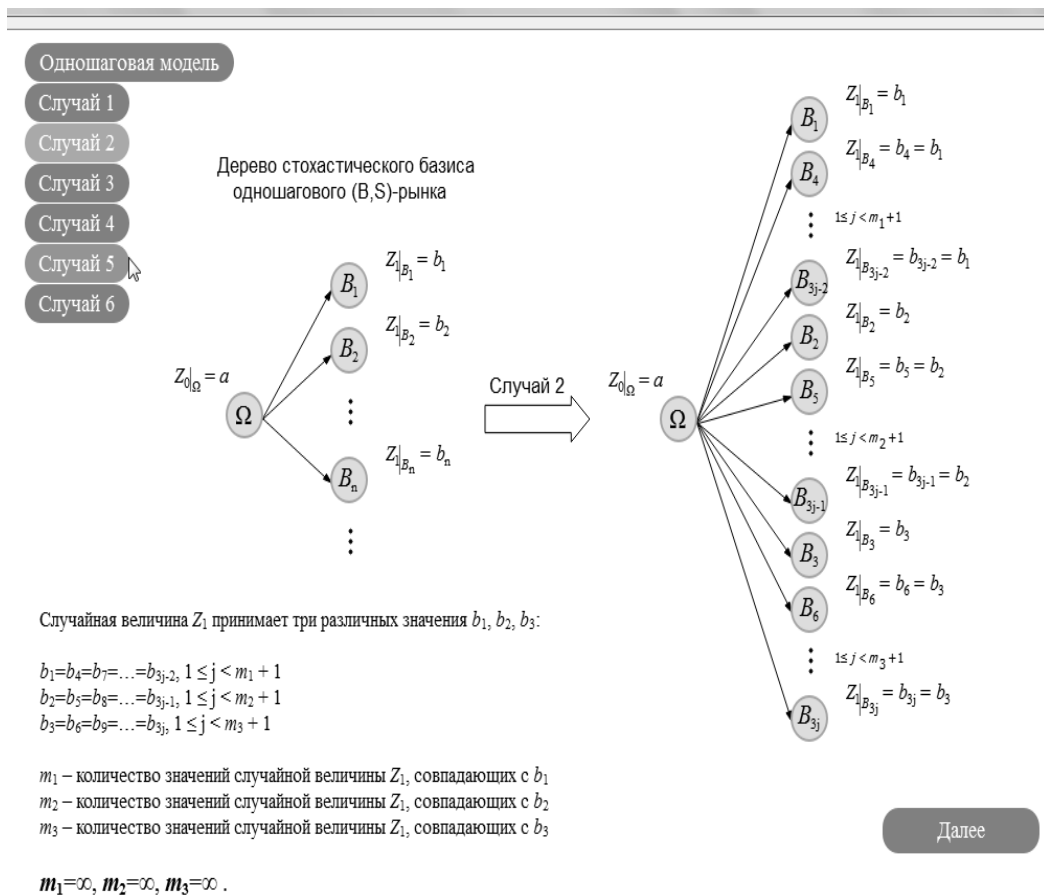


Рис. 27

Одношаговая модель

- Случай 1
- Случай 2
- Случай 3
- Случай 4
- Случай 5
- Случай 6

$$D_1 = \bigcup_{j=1}^{m_1} B_{3j-2} \quad D_2 = \bigcup_{j=1}^{m_2} B_{3j-1} \quad D_3 = \bigcup_{k=1}^{m_3} B_{3k}$$

Параметры

$m_1 =$ $m_2 =$ $m_3 =$

$a =$ $b_1 =$ $b_2 =$ $b_3 =$

$b_1 < a < b_2 < b_3$

Задать платежное обязательство:

цена исполнения =

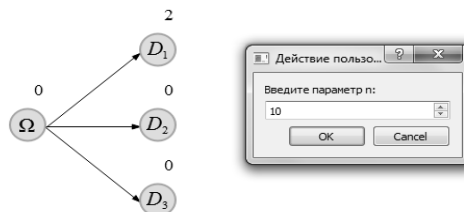
Введите значения случайного процесса Z , задайте тип платёжного обязательства и нажмите "Начать вычисления".

Назад

Начать вычисления

Рис. 28

- обязательства на реплицируемость.
- Шаг 2. Вычисление цены финансового обязательства и компонент мартингалльной меры.
- Шаг 3. Построение специальной хааровской интерполяции.
- Шаг 4. Вычисление компонент самофинансируемого портфеля.
- Шаг 5. Конец.



Пусть $P=(p_1, p_2, \dots, p_n, \dots)$: $p_i = P(B_i) > 0, i=1, 2, 3, \dots$ - вероятностная мера, относительно которой процесс Z является мартингалом. С учетом значений процесса Z , обозначим x_1, x_2, x_3 .

f не реплицируемо на исходном рынке.

Рис. 29

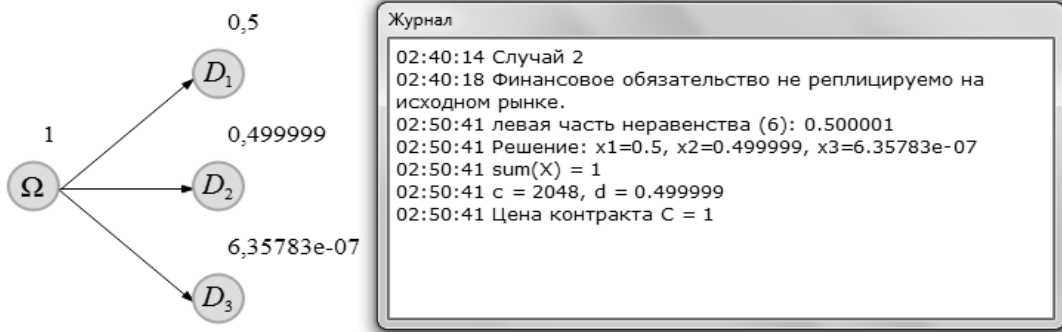
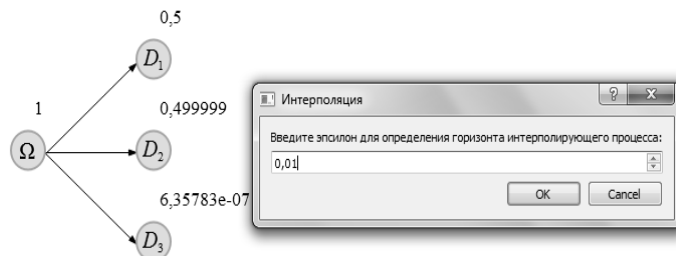


Рис. 30

- мартингальной меры.
- Шаг 3. Построение специальной хааровской интерполяции.
- Шаг 4. Вычисление компонент самофинансируемого портфеля.
- Шаг 5. Конец.



- Алгоритм
- Результаты
- Свойства

Для интерполирующего рынка определяем вычислительный горизонт N . Типичной схемой специальной хааровской фильтрации является схема, когда при переходе от момента времени n к моменту $n+1$, $n=1,2,3,\dots$ дробится только один атом, причём тот, который в результате дробления был получен на предыдущем шаге. Порядок появления событий $B_i, i=1,2,3,\dots$ между моментами времени n и $n+1$, $n=1,2,3,\dots, N$ определяется с помощью генератора случайных векторов, основанного на геометрическом распределении.

Цена контракта: $C=1$
 Мартингальная мера:
 $x1=0.5$
 $x2=0.499999$
 $x3=6.35783e-07$

Рис. 31

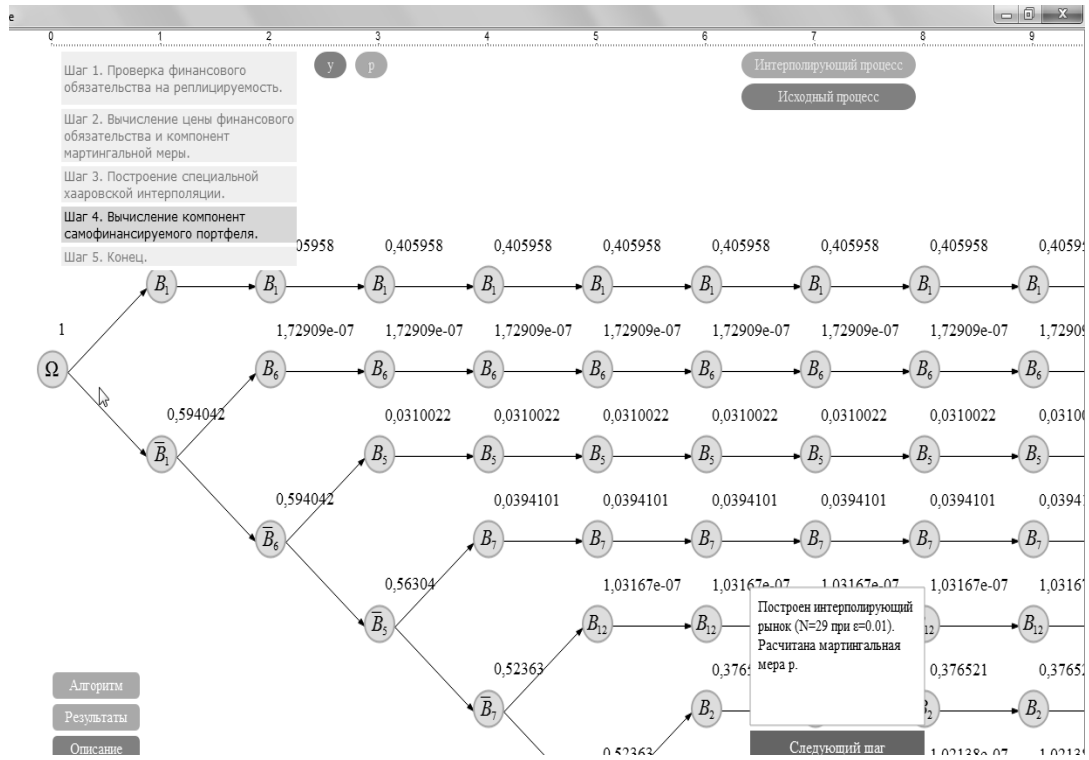


Рис. 32

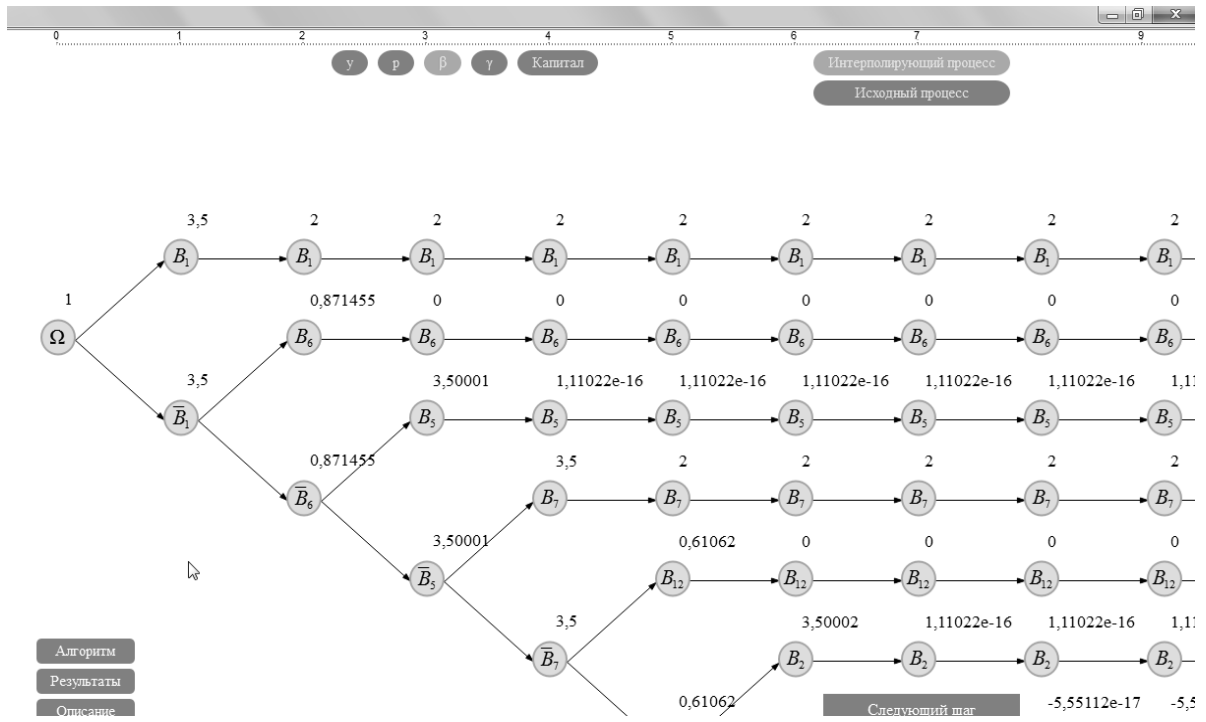


Рис. 33

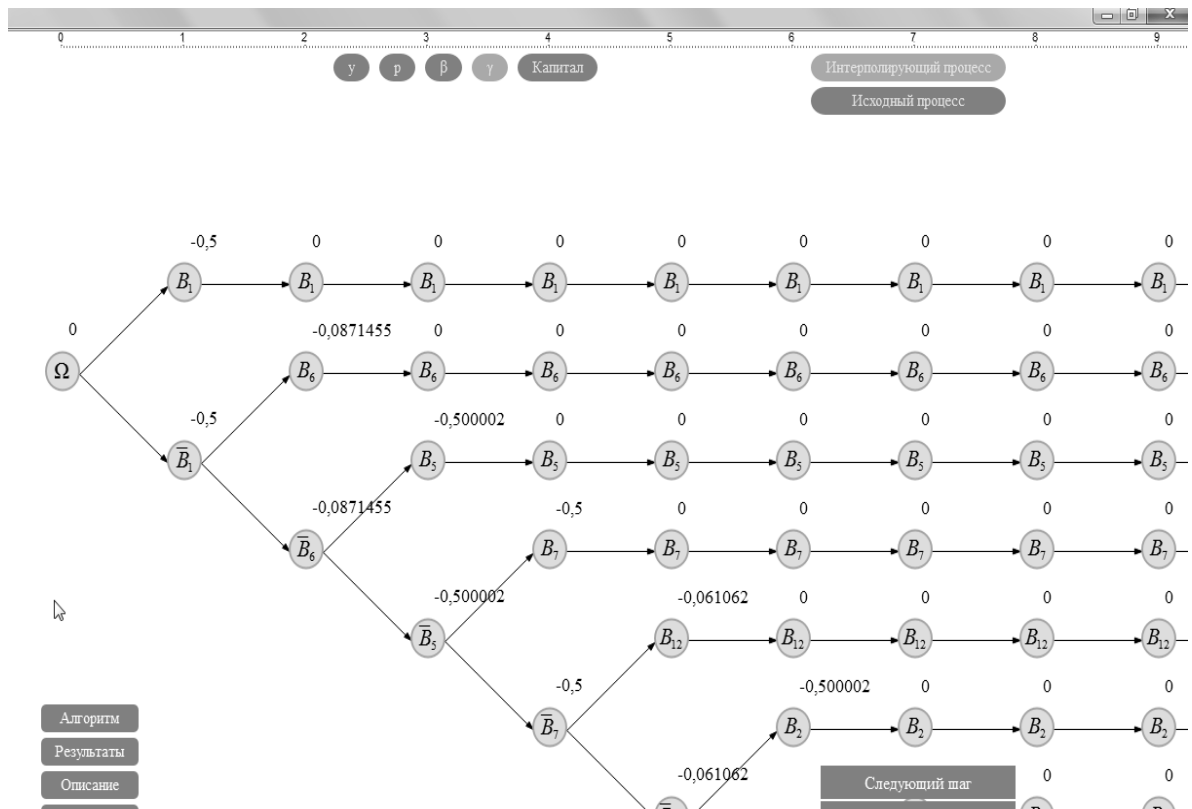


Рис. 34

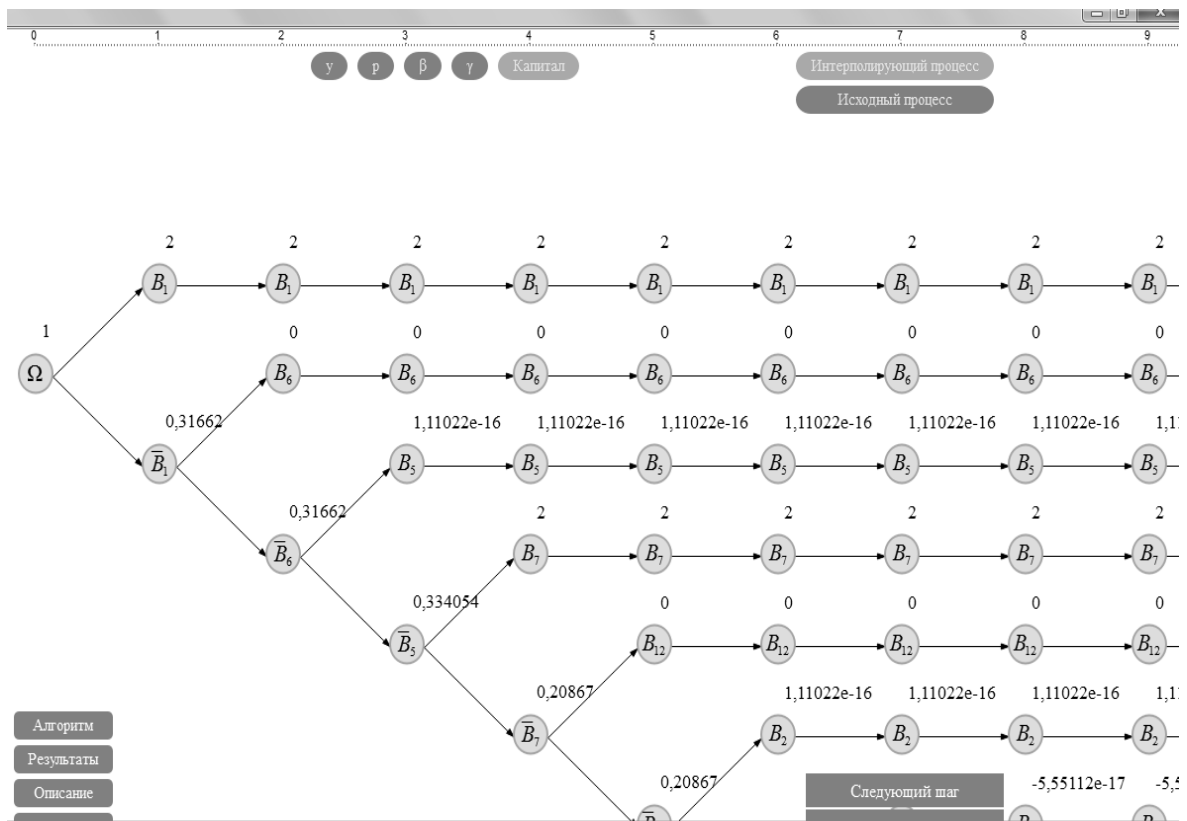


Рис. 35

ЗАКЛЮЧЕНИЕ

В заключение приведем основные результаты нашего исследования, выносимые на защиту.

1. Доказательство существования мартингалльных мер, удовлетворяющих ОСУХЕ, в случае произвольного начального значения цены акции и **трёх** произвольных значений (всевозможных кратностей) финальной цены акции. Создание алгоритма для вычисления этих мартингалльных мер.
2. Доказательство существования мартингалльных мер, удовлетворяющих ОСУХЕ, в случае произвольного начального значения цены акции и **любого конечного числа** рациональных значений (всевозможных кратностей) финальной цены акции. Создание алгоритма для вычисления полученных мартингалльных мер.
3. Доказательство существования мартингалльных мер, удовлетворяющих ОСУХЕ, когда начальное значение цены акции находится в определённом интервале, а значения цены акции в финальный момент времени экспоненциально растут.
4. Новые формулы для вычисления совершенных хеджирующих стратегий на интерполирующем рынке (\mathbf{B}, \mathbf{S}) -рынке с бесконечным горизонтом с использованием мартингалльных мер, удовлетворяющих ОСУХЕ.

5. Разработанные приближённые методы квантильного хеджирования на интерполирующих (\mathbf{B}, \mathbf{S}) -рынках.
6. Новый научно-исследовательский программный комплекс, написанный на языке C++ в среде Qt Creator, фреймворк Qt5, реализующий полученные в диссертации алгоритмы и формулы.

Автором диссертации был исследован целый спектр статических моделей неполных безарбитражных (\mathbf{B}, \mathbf{S}) -рынков со счётным числом состояний. Для всех моделей были получены достаточные условия существования мартингальных мер, удовлетворяющих ОСУХЕ, с помощью которых можно интерполировать заданные неполные рынки до полных. Полученные в диссертации результаты вносят значительный вклад в стохастическую финансовую математику. Алгоритмы программного комплекса и непосредственно программный комплекс облегчают выбор оптимального поведения инвесторов на финансовых рынках.

Литература

- [1] *Ахо А.* Структуры данных и алгоритмы./А. Ахо, Д. Хопкрофт, Д. Ульман// М: Вильямс.- 2000.
- [2] *Белявский Г.И.* Исследование модели (B, S) -рынка относительно специальной хааровской фильтрации./ Г.И. Белявский, В.В. Мисюра, И.В. Павлов // Международная школа-семинар по геометрии и анализу памяти Н.В. Ефимова. Тезисы докладов. Ростов-на-Дону, 1998.- С.179-181.
- [3] *Белявский Г.И.* Ранговый критерий полноты одного финансового рынка при допущении арбитража./ Г.И. Белявский, В.В. Мисюра, И.В. Павлов // Обозрение прикл. и промышл. матем., 1999.- Т.6.- В.1.- С.121-122.
- [4] *Богачев В.И.* Основы теории меры./ В.И. Богачев // Москва-Ижевск: НИЦ Регулярная и хаотическая динамика, Институт компьютерных исследований. Том 1, 2006.-584 с.
- [5] *Богачев В.И.* Действительный и функциональный анализ. / В.И. Богачев, О.Г. Смолянов// М.-Ижевск: НИЦ Регулярная и хаотическая динамика, Институт компьютерных исследований, 2009.-724 с.
- [6] *Богачёва М.Н.* Об интерполяции финансовых рынков в случае конечно вероятностного пространства. / М.Н. Богачёва // Математические и статистические методы в экономике и естествознании:

Материалы 3-х межвузовских научных чтений, Ростов-на-Дону, РГЭУ. 2002.- С.126-128.

- [7] *Богачёва М.Н.* Моделирование безарбитражных финансовых рынков и интерполяционные методы ее исследования. / М.Н. Богачёва // Диссертация на соискание ученой степени кандидата физико-математических наук. Ростов-на-Дону.- 2004.
- [8] *Богачева М.Н.* Критерий существования мартингальной меры, удовлетворяющей свойству хааровской единственности. / М.Н. Богачева, И.В. Павлов // Обозрение прикл. и промышл. матем., М.: ТВП, 2001.- Т.8.- В.2.- С.743-745.
- [9] *Богачева М.Н.* Полное описание мартингальных мер, удовлетворяющих свойству хааровской единственности, для одного финансового рынка. / М.Н. Богачева, И.В. Павлов // Обозрение прикл. и промышл. матем., М.: ТВП, 2001.- Т.8.- В.2.- С.745-747.
- [10] *Богачева М.Н.* О хааровских расширениях безарбитражных финансовых рынков до безарбитражных и полных. / М.Н. Богачева, И.В. Павлов // Успехи матем. наук, 2002.- Т.57.- В.3.- С.143-144.
- [11] *Богачева М.Н.* О хааровских расширениях безарбитражных финансовых рынков до безарбитражных и полных. / М.Н. Богачева, И.В. Павлов // Известия вузов. Северо-Кавказский регион. Естеств. науки. 2002.- №3.- С.16-24.
- [12] *Богачева М.Н.* О моделировании финансовых рынков при наличии двух агрессивных скупщиков акций. / М.Н. Богачева, И.В. Павлов // Математические и статистические методы в экономике и естествознании: Материалы 3-х межвузовских научных чтений, Ростов-на-Дону, РГЭУ. 2002.- С.133-135.

- [13] *Боди З.* Финансы. / З. Боди, Р.К. Мертон // М.:Вильямс.- 2003.-585 с.
- [14] *Бланшет Ж.* Qt 4: Программирование GUI на C++. / Ж. Бланшет, М. Саммерфилд // М.: Кудиц-Пресс.- 2007.
- [15] *Браунси К.* Основные концепции данных и реализация в C++. / К. Браунси // М: Вильямс, 2002.
- [16] *Вадзинский Р.Н.* Справочник по вероятностным распределениям. / Р.Н. Вадзинский // М.:Наука, 2001.-296 с.
- [17] *Васильев Ф.П., Иваницкий А.Ю.* Линейное программирование. / Ф.П. Васильев, А.Ю Иваницкий // М.: Факториал-пресс, 2003.- 348 с.
- [18] *Волков С.Н.* О методологии хеджирования опционов. / С.Н. Волков, Д.О. Крамков // Обозрение прикл. и промышл. матем., М.:ТВП, 1997.- Т.4.- №1.- С.18-65.
- [19] *Волосатова Т.А.* Применение случайных хааровских интерполяций к моделированию (B, S) -рынков с двумя агрессивными скупщиками акций. / Т.А. Волосатова // Обозрение прикл. и промышл. матем., М.: ТВП, 2005.- Т.12.- В.3- С.713-714.
- [20] *Волосатова Т.А.* Модели финансовых рынков, допускающих арбитраж, и их исследование с помощью метода хааровских интерполяций. / Т.А. Волосатова // Диссертация на соискание ученой степени кандидата физико-математических наук. Ростов-на-Дону, 2006.
- [21] *Волосатова Т.А.* Совершенные хеджи в полных финансовых рынках, имитирующих скупку акций и допускающих арбитраж. /

- Т.А. Волосатова, И.В. Павлов // Обозрение прикл. и промышл. матем., М.: ТВП, 2003.- Т.10.- В.2.- С.341-342.
- [22] *Выхристов В.А.* О финансовых расчетах на безарбитражных (B, S) -рынках с конечным числом агрессивных скупщиков акций. / В.А. Выхристов, Г.А. Можаяев // Обозрение прикл. и промышл. матем., 2007.- Т.14.- В.5.- С.769-789.
- [23] *Выхристов В.А.* Примеры мартингалов, не удовлетворяющих ОСУХЕ, и их приближение мартингальными мерами, удовлетворяющими ОСУХЕ. / В.А. Выхристов, Г.А. Можаяев // Обозрение прикладной и промышленной математики, Москва, ТВП. 2007.- Т.14.- В.3.- С.523-524.
- [24] *Данекянц А.Г.* О специальных хааровских интерполяциях мартингалов. / А.Г. Данекянц // Известия вузов. Северо-Кавказский регион. Естественные науки, приложение, 2005.- №3.- С.3–20.
- [25] *Данекянц А.Г.* Моделирование безарбитражных финансовых рынков с помощью хааровских интерполяций на счетном вероятностном пространстве. / А.Г. Данекянц // Диссертация на соискание ученой степени кандидата физико-математических наук. Ростов-на-Дону, 2006.
- [26] *Данекянц А.Г.* Техника интерполяции финансовых рынков, реализованных на счетном вероятностном пространстве. / А.Г. Данекянц, И.В. Павлов // Обозрение прикл. и промышл. матем., М.: ТВП, Т.10, В.2, 2003.- С.345–346.
- [27] *Данекянц А.Г.* Свойства хааровских интерполяций мартингалов в случае потока атомарных σ -алгебр и бесконечного горизонта. /

А.Г. Данекянц, И.В. Павлов // Обзорение прикл. и промышл. матем., М.: ТВП, 2004.- Т.11.- В.1.- С.112-113.

- [28] *Данекянц А.Г.* Модель (B, S) -рынка с бесконечным числом скупщиков акций и совершенное хеджирование методом хааровских интерполяций. / А.Г. Данекянц, И.В. Павлов// Обзорение прикл. и промышл. матем., М.: ТВП, 2005.- Т.12.- В.1.- С.143–144.
- [29] *Данекянц А.Г.* Аппроксимационно-интерполяционный метод сведения безарбитражных финансовых рынков с бесконечным числом состояний к безарбитражным и полным рынкам с конечным числом состояний./ А.Г. Данекянц, И.В. Павлов // Обзорение прикл. и промышл. матем., М.: ТВП, 2005.- Т.12, В.3, С.730–731.
- [30] *Ерёмин И.И.* Введение в теорию линейного и выпуклого программирования. / И.И. Ерёмин, Н.Н. Астафьев // М.: ФИЗМАТЛИТ, 1976.-192 с.
- [31] *Измайлов А.Ф.* Численные методы оптимизации. / А.Ф. Измайлов, М.В. Солодов // М.: ФИЗМАТЛИТ, 2005.
- [32] *Кнут Д.* Искусство программирования. / Д. Кнут // М: Вильямс, 2000.- Т.1,2,3.
- [33] *Кутуков В.Б.* Основы финансовой и страховой математики. / В.Б. Кутуков // М.: Дело, 1998.
- [34] *Липцер Р.Ш.* Теория мартингалов. / Р.Ш. Липцер, А.Н. Ширяев // М.: Наука, 1986.-512 с.
- [35] *Макорин А.* GLPK (GNU Linear Programming Kit) — пакет предназначенный для решения больших задач линейного программирования./ А. Макорин // <http://www.gnu.org/software/glpk>.

- [36] *Малыхин В.И.* Финансовая математика. /В.И. Малыхин// М.:ЮНИТИ, 1999.
- [37] *Мельников А.В.* Финансовые рынки: стохастический анализ и расчет производных ценных бумаг./ А.В. Мельников // М.:ТВП, 1997.-126 с.
- [38] *Мельников А.В.* Математика финансовых обязательств. / А.В. Мельников, С.Н. Волков, М.Л. Нечаев// М.: ГУ ВШЭ, 2001.-254 с.
- [39] *Мельников А.В.* О дискретной модели финансового рынка и методах расчетов с ценными бумагами. / А.В. Мельников, М.Л. Нечаев, В.М. Степанов // Препринт, М.: Научно-иссл. Актуарно-финансовый центр, 1996.- №3.
- [40] *Мельников А.В.* Вопросы безарбитражности и полноты дискретных рынков и расчеты платежных обязательств. / А.В. Мельников, К.М. Феоктистов// Обозрение прикл. и промышл. матем., М.: ТВП, 2001.-Т.8, В.1, С.28-40.
- [41] *Можжаев Г.А.* Случайные интерполяции финансовых рынков с тремя агрессивными скупщиками акций. / Г.А. Можжаев // Изв. вузов. Северо-Кавказский регион. Естеств. науки. 2006.- Приложение №12(48), С.4-17.
- [42] *Можжаев Г.А.* Построение хеджирующих стратегий при наличии трех агрессивных скупщиков акций./ Г.А. Можжаев // Материалы региональной научно-практической конференции профессорско-преподавательского состава, Ростов-на-Дону, 2006.- С.41-43.

- [43] *Можжаев Г.А.* Проверка мартингальной меры на ослабленное свойство универсальной хааровской единственности. / Г.А. Можжаев// Материалы региональной научно-практической конференции профессорско-преподавательского состава, Ростов-на-Дону, 2007.- С.46-47.
- [44] *Можжаев Г.А.* Моделирование финансовых рынков с произвольным числом агрессивных скупщиков акций. / Г.А. Можжаев// Диссертация на соискание ученой степени кандидата физико-математических наук. Ростов-на-Дону, 2007.
- [45] *Можжаев Г.А.* Описание двух алгоритмов, входящих в методику расчетов на безарбитражных финансовых рынках с конечным числом агрессивных скупщиков. / Г.А. Можжаев, В.А. Выхристов // Обзорное прикл. и промышл. матем., М.: ТВП, 2006.- Т.13.- В.6.- С.1037-1038.
- [46] *Можжаев Г.А.* О финансовых расчетах на безарбитражных (B, S) -рынках с бесконечным числом агрессивных скупщиков акций. / Г.А. Можжаев, И.В. Павлов, Э.А. Пилюсян // Обзорное прикл. и промышл. матем., 2008.- Т.15.- В.3.- С.505-506.
- [47] *Назарько О.В.* Слабые деформации на бинарных финансовых рынках./ О.В. Назарько // Известия ВУЗов. Северо-Кавказский регион. Естественные науки, 2010.- В.1.- С.12-18
- [48] Хеджирование опционов с заданной вероятностью. / А.А. Новиков // Теория вероятностей и ее применения, 1998.- Т.43.- №1.- С.152-160.
- [49] *Павлов И.В.* Об одной модели (B, S) -рынка, связанной с простейшей фильтрацией Хаара. / И.В. Павлов // Обзорное прикладной и промышленной математики. 1997, Т.4, В.3, С.389-390.

- [50] Павлов И.В. Обобщенная модель эволюции цен акций в случае их скупки. / И.В. Павлов и др. // Известия РГСУ, 2000.- №5, С.165-173.
- [51] Павлов И.В. Модели (B, S) -рынков типа Кокса-Росса-Рубинштейна в случае скупки акций. / И.В. Павлов и др. // Известия вузов. Северо-Кавказский регион. Естеств. науки, 2001.- №1.- С.7-11.
- [52] Павлов И.В. Приближение мартингалльных мер мартингалльными мерами, удовлетворяющими ослабленному свойству универсальной хааровской единственности. / И.В. Павлов, Г.А. Можяев, В.А. Выхристов // Материалы региональной научно-практической конференции профессорско-преподавательского состава, Ростов-на-Дону, 2007.- С.49-50.
- [53] Павлов И.В. Интерполяция мартингалов относительно хааровских фильтраций. / И.В. Павлов, А.Г. Данекянц // Обзорение прикл. и промышл. матем., М.: ТВП, 2004.- Т.11. В.1- С.73–82.
- [54] Павлов И.В. Об ослабленном свойстве универсальной хааровской единственности. / И.В. Павлов, А.Г. Данекянц // Обзорение прикл. и промышл. матем., М.: ТВП, 2004.- Т.11.- В.3.- С.506-508.
- [55] Павлов И.В. Критерий существования мартингалльной меры и расчёт цены опциона в случае специальной хааровской фильтрации./ И.В. Павлов, В.В. Мисюра // Изв. вузов Северо-Кавказский регион. Естеств. науки. 1998.- №4.- С.24-30.
- [56] Павлов И.В. Методика финансовых расчетов на безарбитражных (B, S) -рынках с конечным числом агрессивных скупщиков акций. / И.В. Павлов, Г.А. Можяев, В.А. Выхристов// Обзорение прикл. и промышл. матем., М.: ТВП, 2006.- Т.13.- В.6.- С.1039-1040.

- [57] *Павлов И.В.* Интерполяционные свойства мартингалльных мер на сепарабельном измеримом пространстве. / И.В. Павлов, Э.А. Пилюсян // *Обзорные прикл. и промышл. матем.*, 2007, Т.14, В.4, С.660-662.
- [58] *Павлов И.В.* Сведение процедуры хеджирования в одношаговой модели (B, S) -рынка со счетным числом состояний к хеджированию на интерполирующем рынке с бесконечным горизонтом. // *Обзорные прикл. и промышл. матем.*, 2009, Т.16, В.4, С.690-691.
- [59] *Пилюсян Э.А.* Алгоритм определения приоритетных, неприоритетных и усредненного скупщика акций в модели агрессивной скупки. / Э.А. Пилюсян // *Обзорные прикл. и промышл. матем.*, 2009.- Т.16.- В.2.- С.375-376.
- [60] *Пилюсян Э.А.* Структура и алгоритмы функционирования программного комплекса «Хеджирование посредством интерполяции». / Э.А. Пилюсян // *Научно-технические ведомости СПбГПУ, "Информатика. Телекоммуникации. Управление."* 2009.- №2.- С. 133-139.
- [61] *Пилюсян Э.А.* Модели и алгоритмы программ для финансовых рынков, подверженных массовой скупке акций. / Э.А. Пилюсян // *Диссертация на соискание ученой степени кандидата технических наук.* Ростов-на-Дону, 2009.
- [62] *Пилюсян Э.А.* Методы финансовых расчетов на безарбитражных (B, S) -рынках с бесконечным числом агрессивных скупщиков акций. / Э.А. Пилюсян, Г.А. Можяев // *Обзорные прикл. и промышл. матем.*, 2008.- Т.15.- В.5.- С.819-834.
- [63] Построение мультисервисных сетей операторов связи на базе технологии Metro Ethernet. // *Электронный ресурс* www.cisco.com/web/RU/downloads/optical.pdf.

- [64] Решения и продукты компании Cisco Systems по построению оптических связей. // Электронный ресурс <http://www.uni.ru/solutions.php?action=show>.
- [65] *Савитч У.* Язык C++. Курс объектно-ориентированного программирования. / У. Савитч // М.: Вильямс, 2001.
- [66] *Фельмер Г.* Введение в стохастические финансы. Дискретное время. / Г. Фельмер, А. Шид // М.: МЦМНО, 2008.- 496 с.
- [67] *Хантер Д.* Введение в XML. / Д. Хантер, К. Кэгл , Д. Гиббонс, Н. Озу, Д. Пиннок, П. Спенсер // М: Лори.- 2001.
- [68] *Чеботарев А.* Библиотека Qt4. Создание прикладных приложений в среде Linux. / А. Чеботарев // М: Вильямс, 2006.
- [69] *Ширяев А.Н.* Вероятность — 1. / А.Н. Ширяев // М.: МЦМНО, 2004.- 520 с.
- [70] *Ширяев А.Н.* Вероятность — 2. / А.Н. Ширяев// М.: МЦМНО, 2004.- 928 с.
- [71] *Ширяев А.Н.* О некоторых понятиях и стохастических моделях финансовой математики. / А.Н. Ширяев // Теория вероятностей и ее применения. 1994.- Т.39- №1.- С.5-22.
- [72] *Ширяев А.Н.* Стохастические проблемы финансовой математики./ А.Н. Ширяев // Обозрение прикл. и промышл. матем., М.:ТВП, 1994.- Т.1.- №5.- С.780-820.
- [73] *Ширяев А.Н.* Основы стохастической финансовой математики. / А.Н. Ширяев // М.: ФАЗИС.- 2004.- Т.1,2.- 1017 с.

- [74] *Ширяев А.Н.* К теории расчетов опционов Европейского и Американского типов. II. Непрерывное время. / А.Н. Ширяев, Ю.М. Кабанов, Д.О. Крамков, А.В. Мельников // Теория вероятностей и ее применения. 1994.- Т.39.- №1. -С.80-129.
- [75] *Шле М.* Qt4. Профессиональное программирование на C++. / М. Шле // СПб.: ВHV-Санкт-Петербург ВHV.- 2007.
- [76] *Эдди С.* XML. Справочник. Наиболее полное руководство. / С. Эдди // СПб.: Питер.- 1999.
- [77] *Long R.* Martingales spaces and inequalities. / R. Long // Peking University Press. 1993. 346 p.
- [78] *Neveu J.* Discrete-parameter martingales. / J. Neveu // North-Holland Publ. Comp., 1975. 236 p.
- [79] *Taqqu M.S.* The analysis of finite security markets. / M.S. Taqqu, W. Willinger // Adv. Appl. Prorab., 1987. 9. p. 1-25.
- [80] *Pavlov I.V.* Construction of the hedging strategies for one model of (B, S) -market. / I.V. Pavlov at al.// Probabilistic Methods in Discrete Math. (Proceedings of the 5-th Intern. Petrozavodsk Conf.), Petr., Russia, Jun, 2000.Utrecht, the Netherlands, 2002. p. 311-316.
- [81] *Shiryaev A.N., Spokoiny V.G.* Statistical experiments and decisions. Asymtotic theory. / A.N. Shiryaev, V.G. Spokoiny // World Scientific Publ. Co. Pte. Ltd., Advanced Series on Statistical Science and Applied Probability, v. 8. 283 p.

Работы автора по теме диссертации

- [82] *Цветкова И.В.* Бесконечномерная задача оптимизации при исследовании финансового рынка с бесконечным числом скупщиков акций.

- / И.В. Цветкова, В.В. Шамраева // Строительство-2011, материалы международной научно-практической конференции.- Ростов-на-Дону.- 2011. С.127-129.
- [83] *Цветкова И.В.* Решение некоторых задач, возникающих при отыскании мартингальной меры в случае финансового рынка с бесконечным числом скупщиков акций. / И.В. Цветкова, В.В. Шамраева // Современные методы и проблемы теории операторов и гармонического анализа и их приложения. Тезисы докладов.- Ростов-на-Дону.- 2011.- С.68-69.
- [84] *Цветкова И.В.* Бесконечномерная задача оптимизации при исследовании финансового рынка со счётным числом состояний. / И.В. Цветкова, В.В. Шамраева // Теория операторов, комплексный анализ и математическое моделирование. Тезисы докладов международной научной конференции (Волгодонск, Россия, 4-8 июля 2011 г.).-Владикавказ: ЮМИ ВНЦ РАН и РСО-А.-2011.- С.179-180.
- [85] *Цветкова И.В.* Некоторые модели финансового рынка с бесконечным числом скупщиков акций. / И.В. Цветкова, В.В. Шамраева // Строительство-2012, материалы международной научно-практической конференции.- Ростов-на-Дону.- 2012.- С 110-112.
- [86] *Павлов И.В.* Некоторые результаты о мартингальных мерах одношаговых моделей финансовых рынков, связанные с условием несовпадения барицентров. / И.В. Павлов, И.В. Цветкова, В.В. Шамраева // Вестник РГУПС.- Ростов-на-Дону.- 2012.- №3.- С.177-181.
- [87] *Павлов И.В.* Достаточные условия существования мартингальной меры, удовлетворяющей ослабленному условию несовпадения барицентров, для финансового рынка со счётным числом состояний. /

И.В. Павлов, И.В. Цветкова, В.В. Шамраева // XX Международная конференция «Математика. Экономика. Образование». Материалы.- Ростов-на-Дону.- 2012.- С.169-170.

[88] *Цветкова И.В.* О множествах мартингалльных мер, удовлетворяющих ослабленному условию несовпадения барицентров (ОУНБ). / И.В. Цветкова, В.В. Шамраева // Современные методы и проблемы теории операторов и гармонического анализа и их приложения-II. Тезисы докладов.- Ростов-на-Дону.- 2012.- С.93-94.

[89] *Цветкова И.В.* Исследование модели финансового рынка с бесконечным числом скупщиков акций с помощью аргументов двойственности. / И.В. Цветкова, В.В. Шамраева // Науковедение. -2012.-№4.- Вып.13. (электронное научное издание, рекомендованное ВАК РФ).

[90] *Павлов И.В.* Сведение хеджирования на неполных рынках с бесконечным числом состояний к хеджированию на полных рынках с бесконечным горизонтом. / И.В. Павлов, И.В. Цветкова, В.В. Шамраева // Международная научная конференция «Современные методы и проблемы теории операторов и гармонического анализа и их приложения – III» , Ростов-на-Дону.-2013.- С.107-108.

[91] *Цветкова И.В.* Расчёт компонентов хеджирующего портфеля с помощью процедуры хааровской интерполяции. / И.В. Цветкова, В.В. Шамраева //Науковедение.- 2013.- №3.- Вып.16. (электронное научное издание, рекомендованное ВАК РФ).

[92] *Цветкова И.В.* Моделирование случайного поведения бесконечного числа скупщиков акций на финансовом рынке. / И.В. Цветкова, В.В. Шамраева //Современные методы прикладной математики, теории управления и компьютерных технологий (ПМТУКТ-2013).

Материалы VI Международной научной конференции.- Воронеж.- 2013.- С.271-273.

- [93] *Павлов И.В.* Хеджирование одношаговых (B, S) - рынков с бесконечным числом состояний с помощью хааровских интерполяций. / И.В. Павлов, И.В. Цветкова, В.В. Шамраева // Обзорение прикладной и промышленной математики.- Москва.- 2013.- Т. 20.- В.2.- С.151-152.
- [94] *Павлов И.В.* Расчёт компонентов хеджирующего портфеля на неполных рынках с недетерминированным поведением скупщиков акций. / И.В. Павлов, И.В. Цветкова, В.В. Шамраева // Инженерный вестник Дона.- 2013.- Вып.4. (электронное научное издание, рекомендованное ВАК РФ).
- [95] *Цветкова И.В.* Новые достаточные условия существования мартингальных мер, удовлетворяющих ОУНБ. / И.В. Цветкова, В.В. Шамраева // Международная научная конференция «Современные методы и проблемы теории операторов и гармонического анализа и их приложения – IV» , Ростов-на-Дону.-2014.- С.133-134
- [96] *Павлов И.В.* О существовании мартингальных мер, удовлетворяющих ослабленному условию несовпадения барицентров: конструктивистский подход. / И.В. Павлов, И.В. Цветкова, В.В. Шамраева // Вестник РГУПС.- Ростов-на-Дону .-2014.- Т.4.- В.56.- С.132-139.
- [97] *Павлов И.В.* О существовании мартингальных мер, удовлетворяющих ослабленному условию несовпадения барицентров./ И.В. Павлов, И.В. Цветкова, В.В. Шамраева // XXV Крымская Осенняя Математическая Школа-симпозиум по спектральным и эволюционным задачам (КРОМШ-2014). Сборник тезисов.- Судак.- 2014.- С. 73.

- [98] *Цветкова И.В.* Программная реализация вычисления компонент хеджирующего портфеля посредством хааровской интерполяции. / И.В. Цветкова // XXVI Крымская Осенняя Математическая Школа-симпозиум по спектральным и эволюционным задачам (КРОМШ-2015). Сборник тезисов.- Симферополь.- 2015.- С.121-122.
- [99] *Цветкова И.В.* Алгоритм вычислительных процедур при построении хеджирующих стратегий на финансовых рынках с бесконечным числом состояний. / И.В. Цветкова // Наукоедение.-2015.-Т.7.-№6. (электронное научное издание, рекомендованное ВАК РФ).
- [100] *Павлов И.В.* О существовании мартингалльных мер, удовлетворяющих ослабленному условию несовпадения барицентров, в случае счётного вероятностного пространства. / И.В. Павлов, И.В. Цветкова, В.В. Шамраева // Теория вероятностей и её применения.- Москва.- 2016.- Т.61.- В.1.- С.173-181.
- [101] *Цветкова И.В.* Алгоритм вычисления мартингалльных мер, удовлетворяющих ОСУХЕ, в случае одношагового рынка со счётным числом состояний. / И.В. Цветкова // Современные методы и проблемы теории операторов и гармонического анализа и их приложения – VI. Тезисы докладов. Ростов-на-Дону.- 2016.- С.144-145.
- [102] *Цветкова И.В.* Алгоритм построения интерполяционных мартингалльных мер в случае счётного вероятностного пространства и конечнозначных цен акций. / И.В. Цветкова // Теория вероятностей и её применения.-Москва.- 2016.- Т.61.- В.3. С.620.
- [103] *Цветкова И.В.* Программный комплекс для хеджирования с помощью интерполяционных мартингалльных мер / И.В. Цветкова // XXVII Крымская Осенняя Математическая Школа-симпозиум по

спектральным и эволюционным задачам (КРОМШ-2016). Сборник тезисов.- Симферополь.- 2016.- С.118.

- [104] *Цветкова И.В.* Свидетельство №2016611994 о государственной регистрации программы для ЭВМ "Построение хеджирующих стратегий посредством хааровских интерполяций" / И.В. Цветкова // Ростовский государственный строительный университет. 2016.

ПРИЛОЖЕНИЕ

В приложении содержится структура исходного кода программного комплекса, описанного в *главе 3* диссертационной работы. А также описание и реализация некоторых классов, интерфейсов и вычислительных алгоритмов.

Структура исходного кода программного комплекса

- application/ (исходный код загрузчика модулей)
- base/ (классы и интерфейсы разработанного фреймворка)
 - |- public/
 - |- rc_plugin_interface.h (листинг А.1.)
 - |- ...
 - |- rc_tree.cpp[h] (листинг А.2.)
 - |- rc_tree_filtration.cpp[h] (листинг А.3.)
 - |- rc_tree_node.cpp[h] (листинг А.4.)
 - |- ...
- plugins/
 - |- research/ (исходный код вычислительного модуля)
 - |- index_object.cpp[h] (листинг А.5. и листинг А.6.)
 - |- model_object.cpp[h] (листинг А.8. и листинг А.9.)
 - |- calculation_dialog.cpp[h] (листинг А.7.)
 - |- ...
- thirdparty/ (сторонние библиотеки)
 - |- glpk/
 - |- qwt-mml/

Листинг А.1. Интерфейс связи основного приложения и плагинов (rc_plugin_interface.h)

```
class RC_BASE_EXPORT RCPluginInterface {
public:
    // уникальный идентификатор модуля
    virtual QString id() = 0;
    // имя модуля
    virtual QString name() = 0;
```

```

    // описание
    virtual QString description () = 0;
    // версия
    virtual QString version () = 0;
    // сведения о создателе модуля
    virtual QString vendor () = 0;

    // эта функция вызывается из основного приложения,
    // чтобы запустить выполнение конкретного модуля
    virtual int execute () = 0;
};

```

```

Q_DECLARE_INTERFACE(RCPluginInterface, "ResearchComplex.PluginInterface")

```

Листинг А.2. Объявление класса, реализующего структуру данных 'дерево', для выполнения расчетов (rc_tree.h)

```

class RC_BASE_EXPORT RCTree : public QObject {
    Q_OBJECT

public:
    explicit RCTree ();
    ~RCTree ();

    // "корень" дерева
    RCTreeNode *root ();

    // глубина дерева (горизонт событий)
    int horizon () const;

    // получить фильтрацию на указанном шаге модели
    RCTreeFiltration *filtration (int step);

    // обновить все узлы дерева в пользовательском интерфейсе
    // (послать сигнал что данные изменились)
    void update ();

private:
    // добавить объект "фильтрация"
    void append (RCTreeFiltration *filtration);
    friend class RCTreeFiltration;

```

```

private:
    // список фильтраций по шагам
    QList<RCTreeFiltration *> filtrations_;

};

```

Листинг А.3. Объявление класса, объекты которого содержат набор атомов (узлов дерева) на конкретном шаге модели (rc_tree_filtration.h)

```

class RCTree;

class RC_BASE_EXPORT RCTreeFiltration : public QObject {
    Q_OBJECT

public:
    // фильтрация всегда принадлежит объекту RCTree
    // и соответствует шагу step модели
    explicit RCTreeFiltration(RCTree *tree, int step);
    ~RCTreeFiltration();

    RCTree *tree();
    // шаг модели
    int step() const;
    // число атомов
    int count() const;
    // получить указатель на узел дерева по индексу
    RCTreeNode *atom(int index);
    // первый атом (верхний)
    RCTreeNode *first();
    // последний атом (нижний)
    RCTreeNode *last();

private:
    void append(RCTreeNode *atom);
    friend class RCTreeNode;

private:
    // список атомов (узлов-дерева)
    QList<RCTreeNode *> atoms_;

    // Указатель на объект-дерево, которому принадлежит данная фильтрация
    RCTree *tree_;

```



```

// Номер шага модели
int step_;
};

```

Листинг А.4. Объявление класса, реализующего узел структуры данных "дерево"(RCTree) (rc_tree_node.h)

```

class RCTree;
class RCTreeFiltration;

class RC_BASE_EXPORT RCTreeNode : public QObject {
    Q_OBJECT

public:
    explicit RCTreeNode(RCTree *tree , RCTreeFiltration *filtration ,
        RCTreeNode *parent); RCTreeNode();

    // удалить переменную на узле дерева
    void remove(const QString &id);

    // получить значение переменной (с идентификатором id) на узле дерева
    QVariant value(const QString &id) const;
    // установить значение переменной (с идентификатором id) на узле дерева
    void setValue(const QString &id , const QVariant &value);

    // получить значение переменной на узле дерева в виде типа double
    double v(const QString &id);
    // установить значение переменной на узле дерева в виде типа double
    void s_v(const QString &id , const double &value);

    // получить значение переменной на узле дерева в виде типа int
    int i(const QString &id);
    // установить значение переменной на узле дерева в виде типа int
    void s_i(const QString &id , const int &value);

    // получить список значений всех переменных на узле
    QVariantList values() const;
    // получить список идентификатором всех переменных на узле
    QStringList ids() const;

    // Указатель на узел-родитель
    RCTreeNode *parent();

```

```

// Получить указатель на узел–потомок (index от 0 до count(), сверху–вниз)
RCTreeNode *child(int index);
// Список узлов–потомков (на которые дробится данный узел)
QList<RCTreeNode *> children();
// Число узлов–потомков
int count() const;

// Послать сигнал что данные изменились
void update();

Q_SIGNALS:
// Сигнал: данные изменились
void changed();

protected:
// Добавить узел в список потомков (с наибольшим индексом)
void append(RCTreeNode *node);

private:
/* map<type, value> */
// хранилище идентификаторов и значений переменных
QMap<QString, QVariant> values_;
// список узлов–потомков
QList<RCTreeNode *> nodes_;

// Указатель на объект–дерево, которому принадлежит данный узел
RCTree *tree_;
// Указатель на объект–фильтрация, которому принадлежит данный узел
RCTreeFiltration *filtration_;
// Указатель на узел–родитель
RCTreeNode *parent_;
};

```

Листинг А.5. Объявление класса, который реализует вычисление компонент мартигальной меры в промежуточные моменты времени (IndexObject.h)

```

class IndexObject {
public:
// m = -1 — означает, что m — бесконечное число
IndexObject(int group, int m, double X);
~IndexObject();

```

```

// Группа индексов. Соответствует одному из m.
// Например, если компоненты меры считаем для m1, то 0;
// если для m2, то 1, ...
int group() const;
//
double X() const { return X_; };
double sumP() const { return sum_p_; };
bool addNewIndex(int index, double *p_out);
bool indexExists(int index);
bool isInfinite() const { return inf_; };
// использовать только в случае isInfinite() == false
bool isFinished() const { return finished_; };
// все посчитанные величины вероятности
QMap<int, double> &p_map();

// статические функции для работы с несколькими
// группами индексов одновременно

// инициализация
static void Initialize(int groups_number /* сколько разных m? */,
                      bool special_case = false /* true только в 1-м случае */);
// добавить объект отвечающий за вычисления в рамках конкретной группы индексов
static void AddObject(IndexObject *obj);
// получить номер группы по произвольному индексу
static int GetSubindex(int index);
// true, если такой индекс существует (зависит от m)
static bool IndexExists(int index);
// добавить новый индекс и посчитать для него компоненту меры
static bool AddNewIndex(int index, double *p_out);
// вернуть все посчитанные индексы
static QList<int> GetIndexes();
// вернуть посчитанную компоненту меры для индекса
static double GetMeasure(int index,
                        bool *exists = 0 /* false если компоненты нет */);
// завершение работы, освобождение памяти
static void Finalize();

private:
// количество групп индексов
static int g_groups_number_;

```

```

// сдвиг между индексами одной группы
static int g_iteration_shift_;
// список рассчитанных индексов (у всех объектов)
static QList<int> g_indexes_list_;
// список объектов IndexObject
static QList<IndexObject *> g_objects_;
// true только для первого случая
static bool g_special_case_;

// группа индексов
int group_;
// m1,m2, ...
int m_;
// величина меры X из которой будем получать компоненты меры p
double X_;
// true если m – бесконечность
bool inf_;

// остаток от X
double sum_p_;

// максимальный индекс для которого в
// данной группе посчитаны компоненты меры p
int max_index_;
// максимальный индекс, который существует
// для данной группы (если m – бесконечная, то не используется)
int max_limit_;

// true – индексы в группе закончились,
// больше нечего вычислять (т.е. все компоненты меры p посчитаны)
bool finished_;

// индексы для которых компоненты меры вычислены
QSet<int> indexes_;
// полученные компоненты меры
QMap<int, double> p_map_;
};

```

Листинг А.6. Реализация класса для вычисления компонент мартингальной меры в промежуточные моменты времени (IndexObject.cpp)

```

int IndexObject::g_groups_number_ = 0;

```

```

int IndexObject::g_iteration_shift_ = 0;
QList<int> IndexObject::g_indexes_list_;
QList<IndexObject *> IndexObject::g_objects_;

bool IndexObject::g_special_case_ = false;

IndexObject::IndexObject(int group, int m, double X)
    : group_(group)
    , m_(m)
    , X_(X)
    , inf_(m == -1)
    , max_index_(0) {
    sum_p_ = 0.0;

    if (!inf_)
        max_limit_ = g_iteration_shift_ * m_ + (group_ + 1);
}

// static
void IndexObject::Initialize(int groups_number, bool special_case) {
    g_groups_number_ = groups_number;
    g_special_case_ = special_case;

    if (g_special_case_) {
        g_iteration_shift_ = 2;
    } else {
        g_iteration_shift_ = g_groups_number_;
    }
}

// static
void IndexObject::AddObject(IndexObject *obj) {
    g_objects_.append(obj);
}

// static
void IndexObject::Finalize() {
    for (int i=0; i < g_objects_.size(); i++)
        delete g_objects_[i];

    g_objects_.clear();
    g_indexes_list_.clear();
}

```

```

// static
bool IndexObject::IndexExists(int index) {
    for (int i=0; i < g_objects_.size(); i++) {
        if (g_objects_[i]->indexExists(index)) {
            return true;
        }
    }

    return false;
}

// static
bool IndexObject::AddNewIndex(int index, double *p_out) {
    if (g_indexes_list_.indexOf(index)>=0) {
        return false;
    }
    for (int i=0; i < g_objects_.size(); i++) {
        if (g_objects_[i]->indexExists(index)) {
            if (g_objects_[i]->addNewIndex(index, p_out)){
                g_indexes_list_.append(index);
                return true;
            }
            break;
        }
    }
    return false;
}

// static
QList<int> IndexObject::GetIndexes() {
    return g_indexes_list_;
}

// static
double IndexObject::GetMeasure(int index, bool *exists) {
    for (int i=0; i < g_objects_.size(); i++) {
        if (g_objects_[i]->p_map().contains(index)) {
            if (exists) {
                *exists = true;
            }
            return g_objects_[i]->p_map().value(index);
        }
    }
}

```

```

    if (exists) {
        *exists = false;
    }

    return 0.0;
}
// static
int IndexObject::GetSubindex(int index)
{
    if (g_special_case_) {
        if (index == 1) {
            return 0;
        }
        return (index % 2) + 1;
    }
    int sub_idx = (index % g_groups_number_);
    if (sub_idx == 0) {
        return (g_groups_number_ - 1);
    }
    return (sub_idx - 1);
}

// public
bool IndexObject::indexExists(int index) {
    if (g_special_case_) {
        return (group_ == IndexObject::GetSubindex(index));
    }
    int sub_idx = IndexObject::GetSubindex(index);
    if (sub_idx == group_) {
        if (inf_) return true;
        int j = (index - group_) / g_groups_number_ + 1;
        return (j < (m_ + 1));
    }
    // для этой группы индекс не существует
    return false;
}

bool IndexObject::addNewIndex(int index, double *p_out)
{
    if (!indexExists(index)) {
        return false;
    }
}

```

```

}
if (indexes_.contains(index)) {
    return false;
}
if (p_map_.contains(index)) {
    indexes_.insert(index);
    double p = p_map_.value(index);
    if (p_out)
    {
        *p_out = p;
    }
    sum_p_ += p;
    return true;
}
// рассчитываем вероятности от p_max_index до p_index
// поскольку по-алгоритмы расчет идет по-порядку
// для p1, p2, p3, ..., p_index
if (index <= max_index_) {
    return false;
}
if (max_index_ == 0) {
    max_index_ = group_ + 1;
} else {
    max_index_ = max_index_ + g_iteration_shift_;
}

for (int i=max_index_; i <= index; i += g_iteration_shift_) {
    if (!inf_) {
        int j = i + g_iteration_shift_;
        if (j >= max_limit_) {
            finished_ = true;
            p_map_.insert(i, X_);
            X_ = 0.0;
            break;
        }
    }
}

double random10 = ((double) grand() / (double)RAND_MAX); // 0 .. 1.0
double p = random10 * X_;
double q = X_ - p;
X_ = q;

```



```

        setNextStepButtonEnabled( false );
    } else {
        setNextStepButtonEnabled( true );
        if ( model_ -> case_type () == 6 )
        {
            setResultsHtml( resultsHtml( "XRCalculated" ) );
        } else {
            setResultsHtml( resultsHtml( "X123Calculated" ) );
        }
    }
} else
if ( step == 3 ) {
    removeScheme( SCHEME_HAAR_TREE_ID );
    if ( model_ -> calculationStep3 () ) {
        addScheme( model_ -> haarScheme () );
        setCurrentScheme( model_ -> haarScheme() -> id () );
        setResultsHtml( resultsHtml( "InterpolationConstructed" ) );
        setCalculateButtonEnabled( false );
        setNextStepButtonEnabled( true );
    }
} else
if ( step == 4 ) {
    model_ -> calculationStep4 ();
    setNextStepButtonEnabled( true );
    setResultsHtml( resultsHtml( "InterpolationPortfolio" ) );
    showMessage( trUtf8( "Компоненты портфеля"
        "&nbsp;&nbsp;&nbsp;<i>#960;</i>&nbsp;&nbsp;&nbsp;вычислены." ) );
}

RCCalculationDialog::calculateStep( step );
}
...

```

Листинг А.8. Определение класса, реализующего вычислительные алгоритмы модели (model_object.h)

```

// schemes
#define SCHEME_ORIGIN_TREE_ID    "OneStepModel_Scheme_Case16_Origin"
#define SCHEME_HAAR_TREE_ID      "OneStepModel_Scheme_Case16_Haar"

class RCTree;
class RCTreeNode;

```

```

class RCTreeFiltration;
class RCScheme;
class ModelObject : public QObject {
Q_OBJECT

public:
    ModelObject(QObject *parent = 0);
    ~ModelObject ();
    // инициализация переменных, общих для случаев 1–5
    void initCase15Common(double a, double b1, double b2, double b3);
    // инициализация значений переменных для каждого из случаев (1–5)
    void initCase1 ();
    void initCase2 ();
    void initCase3(int m2);
    void initCase4(int m1);
    void initCase5(int m3);
    // инициализация данных для случая 6
    void initCaseR(int r, double a, QList<double> &b, QList<int> m);
    // установка платежного обязательства
    void setPayoff(int type /* тип – Put или Call*/,
                  double strikePrice /* цена исполнения */);
    RCTree *oneStepTree ();
    RCScheme *oneStepScheme ();
    RCTree *specialHaarTree(int horizon);
    RCScheme *haarScheme ();
    RCLog *logObject ();
    // каждая функция – отдельный шаг общего алгоритма
    void calculationStep0 ();
    bool calculationStep1 ();
    bool calculationStep2 ();
    bool calculationStep3 ();
    void calculationStep4 ();
    // цена контракта
    double contractPrice ();
    // получить значения X1,X2,X3 (для случаев 1–5)
    void getX123(double *x1, double *x2, double *x3);
    // получить значения X1,X2,...Xr (случай 6)
    QList<double> getXR ();
    // текст проблемы, если вычисления не были завершены
    QString lastErrorMessage ();
    // значение эpsilon (при вычислении горизонта интерполяции N)

```

```
double epsilon() const;
int case_type() const;
```

```
private:
```

```
// создает объект RSCHEME (визуальное отображение дерева,
// на котором ведутся расчеты)
RCScheme *schemeFromTree(const QString &id,
    RCTree *tree, const QString &symbolA, const QString &symbolB);

// реплицируемо ли платежное обязательство?
bool isPayoffReplicate(RCTree *tree, double &fairPrice);

// возвращает значение случайной величины (геометр. распределение)
int GeometricDistribution(double p);
// решает систему методом Гауса
QList<int> gaussSolve(double **a, int m, int n);
// для конкретного узла дерева подготавливает данные
// (в функцию gaussSolve)
void solveMatrixForNode(RCTreeNode *atom);
bool oneStepSolve(RCTreeNode *p_atom, int goDir);

// Вычисление X1,X2,X3 (случай 1)
void findX123_case1();
// шаг 2 расчетов (случай 1)
bool calculationStep2_case1();
// шаг 2 расчетов (случай 2-3)
void calculationStep2_case23();
// проверка условий для n (случай 2-3)
bool isNArgGood_case23();
// Вычисление X1,X2,X3 (случай 2-3)
void findX123_case23();
// шаг 2 расчетов (случай 4)
void calculationStep2_case4();
// проверка условий для n (случай 4)
bool isNArgGood_case4();
// Вычисление X1,X2,X3 (случай 4)
void findX123_case4();
// шаг 2 расчетов (случай 5)
void calculationStep2_case5();
// проверка условий для n (случай 5)
bool isNArgGood_case5();
```

```
// Вычисление X1,X2,X3 (случай 5)
void findX123_case5 ();
// шаг 2 расчетов (случай 6)
bool calculationStep2_case6 ();
// шаг 3 расчетов (случай 1)
bool calculationStep3_case1 ();
// шаг 3 расчетов (случай 2–5)
bool calculationStep3_case25 ();
// шаг 3 расчетов (случай 6)
bool calculationStep3_case6 ();

// исходные параметры
int case_type_; // случай
int m1_; // значение m1 (отрицательное означает бесконечность)
int m2_; // значение m2 (отрицательное означает бесконечность)
int m3_; // значение m3 (отрицательное означает бесконечность)
///  

// глобальные переменные

// для случаев 2–5
int n_;

// цена контракта
double c_;
double c_min_;
double c_max_;

// временные переменные
double c_tmp_;
double epsilon_;

// для случая 6
double a_; // значение в нулевой момент времени
QList<double> b_; // список значений на атомах В
QList<int> m_; // список значений m
int r_; // число разных m
QList<double> x_;

// структура для расчетов на исходном процессе
RCTree *tree_;
```

```

// структура для расчетов на интерполирующем процессе
RCTree *haarTree_;
// визуальная схема дерева для исходного процесса
RCScheme *scheme_;
// визуальная схема дерева для интерполирующего процесса
RCScheme *haarScheme_;
RCLog *log_;

QString lastErrorMessage_;
};

```

Листинг А.9. Реализация основных вычислительных процедур на примере случая 6
(model_object.cpp)

```

#define VAL_Z      "Z_VALUE_ID"
#define VAL_VAR_F  "F_VAR_VALUE_ID"
#define VAL_F      "F_VALUE_ID"
#define VAL_X      "X_VALUE_ID"

// компоненты меры p
#define VAL_P      "P_VALUE_ID"

// компоненты портфеля
#define VAL_GAMMA  "GAMMA_VALUE_ID"
#define VAL_BETA   "BETA_VALUE_ID"
#define VAL_CAPITAL "CAPITAL_VALUE_ID"

// индекс атома
#define VAL_IDX     "INDEX_VALUE_ID"

...

void ModelObject::calculationStep0()
{
    // подготовка к вычислениям
    // заполнение временных переменных
    c_tmp_ = 0.00;
    c_min_ = 0.00;
    c_max_ = 0.00;

    RCTreeNode *root = tree_>root();
    root->s_v(VAL_VAR_F, 0.0);
}

```

```

..
for (int k=0; k <= tree->horizon(); k++) {
    RCTreeFiltration *f = tree->filtration(k);
    for (int j=0; j < f->count(); j++) {
        RCTreeNode *atom = f->atom(j);
        atom->s_v(VAL_VAR_F, atom->v(VAL_F));
        ...
    }
}
}
...
// проверка платежного обязательства на реплицируемость
bool ModelObject::isPayoffReplicate(RCTree *tree, double &fairPrice) {
    for (int k = tree->horizon() - 1; k >= 0; k--) {
        RCTreeFiltration *f = tree->filtration(k);
        RCTreeNode *p_atom = 0;

        for (int i=0; i < f->count(); i++) {
            p_atom = f->atom(i);

            if (p_atom->count() > 1) {
                p_atom = f->atom(i);

                RCTreeNode *atom_1 = p_atom->child(0);
                RCTreeNode *atom_2 = 0;

                bool diff = false;
                for (int j = 0; j < p_atom->count(); j++) {
                    atom_2 = p_atom->child(1);
                    if (atom_1->v(VAL_Z) != atom_2->v(VAL_Z)) {
                        diff = true;
                        break;
                    }
                }
            }

            if (diff) {
                p_atom->s_v(VAL_GAMMA,
                    (atom_1->v(VAL_VAR_F) - atom_2->v(VAL_VAR_F))
                    / (atom_1->v(VAL_Z) - atom_2->v(VAL_Z)));
                p_atom->s_v(VAL_BETA, atom_1->v(VAL_VAR_F)
                    - p_atom->v(VAL_GAMMA) * atom_1->v(VAL_Z));
            }
        }
    }
}

```

```

p_atom->s_v(VAL_VAR_F, p_atom->v(VAL_Z)
    * p_atom->v(VAL_GAMMA) + p_atom->v(VAL_BETA));

for (int j = 0; j < p_atom->count(); j++) {
    RCTreeNode *atom = p_atom->child(j);
    atom->s_v(VAL_BETA, p_atom->v(VAL_BETA));
    atom->s_v(VAL_GAMMA, p_atom->v(VAL_GAMMA));
}

for (int j=2; j < p_atom->count(); j++) {
    RCTreeNode *atom = p_atom->child(j);
    if (atom != atom_2) {
        if (qAbs(qAbs(atom->v(VAL_Z) * p_atom->v(VAL_GAMMA)
            + p_atom->v(VAL_BETA))
            -qAbs(atom->v(VAL_VAR_F))) > 10E-12) {
            return false;
        }
    }
}
} else {
    return false;
}
} else
if (p_atom->count() == 1) {
    RCTreeNode *atom_1 = p_atom->child(0);

    if (k == (tree->horizon() - 1)) {
        atom_1->s_v(VAL_GAMMA, 0.0);
        atom_1->s_v(VAL_BETA, atom_1->v(VAL_F));
        atom_1->s_v(VAL_VAR_F, atom_1->v(VAL_F));
    }

    p_atom->s_v(VAL_GAMMA, atom_1->v(VAL_GAMMA));
    p_atom->s_v(VAL_BETA, atom_1->v(VAL_BETA));
    p_atom->s_v(VAL_VAR_F, atom_1->v(VAL_VAR_F));

}
}
}

```

```
// вычислим полный капитал портфеля
```



```

tree->root()->s_v(VAL_CAPITAL,
    tree->root()->v(VAL_GAMMA) * tree->root()->v(VAL_Z)
    + tree->root()->v(VAL_BETA));
tree->root()->s_v(VAL_GAMMA, 0.0);
tree->root()->s_v(VAL_BETA, tree->root()->v(VAL_CAPITAL));

fairPrice = tree->root()->v(VAL_CAPITAL);

for (int n=1; n <= tree->horizon(); n++) {
    RCTreeFiltration *f = tree->filtration(n);
    for (int j=0; j < f->count(); j++) {
        RCTreeNode *atom = f->atom(j);
        atom->s_v(VAL_CAPITAL, atom->v(VAL_BETA)
            + atom->v(VAL_GAMMA) * atom->v(VAL_Z));
    }
}

return true;
}
...
bool ModelObject::calculationStep1()
{
    calculationStep0();

    double fair_price = 0.00;
    if (isPayoffReplicate(tree_, fair_price)) {
        RCMessage(trUtf8("Финансовое_обязательство_"
            "реплицируемо_на_исходном_рынке, C=%l")
            .arg(fair_price));
        c_ = fair_price;
        RCSchemeDisplayPropertyGroup *grp = scheme_->displayPropertyGroup();
        grp->addProperty(VAL_BETA, SYMB_BETA, SYMB_BETA);
        grp->addProperty(VAL_GAMMA, SYMB_GAMMA, SYMB_GAMMA);
        grp->addProperty(VAL_CAPITAL, trUtf8("Капитал"), trUtf8("Капитал"));
        scheme_->updateSchemeFromTree();
        grp->update();

        return true;
    } else {
        RCSchemeDisplayPropertyGroup *grp = scheme_->displayPropertyGroup();
        grp->removeProperty(VAL_BETA);
    }
}

```

```

    grp->removeProperty (VAL_GAMMA);
    grp->removeProperty (VAL_CAPITAL);

    RCMessage (trUtf8 ("Финансовое_обязательство_"
                      "не_реплицируемо_на_исходном_рынке."));
}
return false;
}
...
// вычисляем компоненты мартингальной меры Xi на атомах Di (i=1,...,r)
bool ModelObject::calculationStep2_case6 ()
{
    x_.clear ();
    for (int i=0; i < r_; i++)
    {
        x_.append (0.0);
    }

    // обновим значения a из дерева
    a_ = tree_->root()->v(VAL_Z);

    // обновим значения b из дерева
    b_.clear ();
    RCTreeFiltration *f = tree_->filtration (1);
    for (int j=0; j < f->count (); j++) {
        RCTreeNode *atom = f->atom (j);
        b_.append (atom->v(VAL_Z));
    }

    // найдем число l
    bool l_found = false;

    int l;
    for (l=0; l < (r_-1); l++)
    {
        if ((b_.at (l) < a_) && (b_.at (l+1) > a_))
        {
            l_found = true;
            break;
        }
    }
}

```

```

if (!l_found)
{ // мы не должны сюда попасть ,
  // поскольку в диалоге идет проверка вводимых параметров
  lastErrorMessage_ = trUtf8("Не_удалось_найти_такое_число_l_,"
                              "что_выполняется_b_l < a < b_l+1.");
  RCMessage(lastErrorMessage_);
  return false;
}

RCMessage(trUtf8("l=%1").arg(l+1));
if ((m_.at(1) == -1) && (m_.at(l+1) == -1))
{ // вариант 1
  int max_n = 0;
  for (int i=0; i < r_; i++)
  {
    double up = 0.0;
    double down = 0.0;
    for (int k=0; k < r_; k++)
    {
      if (k == 1) continue;
      if (k == (l+1)) continue;
      up += qAbs(b_.at(i) - b_.at(k));
    }
    down = qAbs(b_.at(i) - a_);

    int n = qFloor(up / down);
    if (n > max_n) max_n = n;
  }
  max_n += 1;
  RCMessage(trUtf8("n=%1").arg(max_n));

  double tilda_one = 1 - ((double)(r_ - 2) / (double)max_n);
  double tilda_b_l = tilda_one * b_.at(l);
  double tilda_b_l_plus1 = tilda_one * b_.at(l+1);

  double sum_bk=0.0;
  for (int k=0; k < r_; k++)
  {
    if (k == 1) continue;
    if (k == (l+1)) continue;

```

```

    sum_bk += b_.at(k);
}
double a_tilda = a_ - (1.0 / (double)max_n) * sum_bk;

for (int k=0; k < r_; k++)
{
    if (k == 1) continue;
    if (k == (l+1)) continue;

    x_[k] = 1.0 / (double) max_n;
}

x_[l] = (tilda_b_l_plus1 - a_tilda) / (b_.at(l + 1) - b_.at(l));
x_[l + 1] = (a_tilda - tilda_b_l) / (b_.at(l + 1) - b_.at(l));
} else
{ // вариант 2
    int max_t = 0;
    for (int i=0; i < r_; i++)
    {
        double up = 0.0;
        double down = 0.0;
        for (int k=0; k < r_; k++)
        {
            if (k == 1) continue;
            if (k == (l+1)) continue;
            up += qAbs(b_.at(i) - b_.at(k));
        }
        down = qAbs(b_.at(i) - a_);

        int t = qFloor(up / down);
        if (t > max_t) max_t = t;
    }
    RCMMessage(trUtf8("t=%1").arg(max_t));
    double n = (double)(max_t * max_t) + 0.5;
    RCMMessage(trUtf8("n=%1").arg(n));

    QList<int> indexes;
    for (int k=0; k < r_; k++)
    {
        if (k == 1) continue;
        if (k == (l+1)) continue;

```

```

    indexes.append(k);
}

int rnd = qFloor(((double) grand() / (double)RAND_MAX)
                * (indexes.count() - 1));
int spec_idx = indexes.at(rnd);
double tilda_one = 1 - ((double)(r_ - 3.0) / (double)n) - (1.0 / sqrt(n));
double tilda_b_l = tilda_one * b_.at(1);
double tilda_b_l_plus1 = tilda_one * b_.at(1+1);

double sum_bk=0.0;
for (int k=0; k < r_; k++)
{
    if (k == 1) continue;
    if (k == (1+1)) continue;
    if (k == spec_idx) continue;
    sum_bk += b_.at(k);
}
double a_tilda = a_ - (1.0 / (double)n) * sum_bk - (1.0 / sqrt(n)) * b_.at(spec_idx);

for (int k=0; k < r_; k++)
{
    if (k == 1) continue;
    if (k == (1+1)) continue;
    if (k == spec_idx) continue;

    x_[k] = 1.0 / (double) n;
}

x_[spec_idx] = 1.0 / sqrt(n);
x_[1] = (tilda_b_l_plus1 - a_tilda) / (b_.at(1 + 1) - b_.at(1));
x_[1 + 1] = (a_tilda - tilda_b_l) / (b_.at(1 + 1) - b_.at(1));
}

double sum_X = 0.0;
c_ = 0.0;
for (int j=0; j < f->count(); j++) {
    RCTreeNode *atom = f->atom(j);
    atom->s_v(VAL_X, x_.at(j));
    sum_X += x_.at(j);
}

```

```

    RCMessage( trUtf8 ("x%1=%2").arg(j+1).arg(x_.at(j)));

    c_ += atom->v(VAL_X) * atom->v(VAL_F);
}

tree_->root()->s_v(VAL_X, sum_X);

RCMessage( trUtf8 ("Цена_контракта_C_=%1").arg(c_));
}
...
bool ModelObject::calculationStep2()
{
    bool success = false;
    RCSchemeDisplayPropertyGroup *grp = scheme_->displayPropertyGroup();
    if (case_type_ == 6)
    {
        success = calculationStep2_case6();
    }
    else {
        ... // случаи 1-5
    }

    if (success) {
        grp->addProperty(VAL_X, "x", "x");
        scheme_->updateSchemeFromTree();
        grp->update();
    }
    return success;
}
...
// конструируем хааровскую интерполяцию
bool ModelObject::calculationStep3_case6()
{
    bool ok;
    epsilon_ = QDialog::getDouble(widget_,
        trUtf8 ("Интерполяция"),
        trUtf8 ("Введите_эпсилон_для_определения_"
            "горизонта_интерполирующего_процесса:"),
        0.1, 0.0000000001, 0.999999999, 5, &ok);
    if (!ok)
        return false;
}

```

```

RCMessage(QString("Эпсилон_=%1").arg(epsilon_));

if (haarTree_) {
    delete haarTree_;
    haarTree_ = NULL;
}

b_.clear();
x_.clear();

QList<double> dList;
{
    RCTreeFiltration *f = tree_>filtration(1);
    for (int j=0; j < f->count(); j++) {
        RCTreeNode *atom = f->atom(j);
        b_.append(atom->v(VAL_Z));
        x_.append(atom->v(VAL_X));
        dList.append(atom->v(VAL_F));
    }
}

IndexObject::Initialize(r_);
for (int i=0; i < r_; i++) {
    IndexObject::AddObject(new IndexObject(i, m_.at(i), x_.at(i)));
}

double maxP = 1.0 - epsilon_;
int idx = 0;
int horizon = 0;

qrand(QDateTime::currentMSecsSinceEpoch());

const int safe_counter_limit = 10000;
int safe_counter = safe_counter_limit;
double sum_P = 0.0;
while (sum_P <= maxP)
{
    horizon++;
    double p_out = 0.0;
    while (true)

```

```

{
    safe_counter--;

    if (safe_counter <= 0) {
        RCMMessage(
            trUtf8("Число_итераций_превысило_%1,"
                "исполнение_алгоритма_остановлено. _Попробуйте_ввести_"
                "большее_значение_эпсилон_и/или_повторить_расчет.")
            .arg(safe_counter_limit));
        IndexObject::Finalize();
        return false;
    }

    idx = GeometricDistribution(1.0 / (horizon + 1.0));
    if (IndexObject::AddNewIndex(idx, &p_out)) {
        break;
    }
};

sum_P += p_out;

RCMessage(QString("A_%1:_p(A_%1)=%2_"
    "(sum(P)_delta_i_=%3)")
    .arg(idx).arg(p_out).arg(sum_P));
}
RCMessage(QString("N_=%1").arg(horizon));
QList<int> indexes = IndexObject::GetIndexes();
QList<int> sortedIndexes(indexes);
qSort(sortedIndexes);
int maxIndex = sortedIndexes.last();
for (int i=1; i<=maxIndex; i++)
{
    bool p_exists;
    double p_out = IndexObject::GetMeasure(i, &p_exists);
    if (p_exists)
    {
        RCMMessage(QString("P[%1]_=%2")
            .arg(i).arg(p_out));
    }
}
}

haarTree_ = specialHaarTree(horizon);

```



```

a_ = tree_>root()->v(VAL_Z);
haarTree_>root()->s_v(VAL_Z, a_);

int atom_index = 0;
for (int i=1; i <= haarTree_>horizon(); i++) {
  RCTreeFiltration *f = haarTree_>filtration(i);
  for (int j=0; j < f->count(); j++) {
    atom_index = indexes.at(i - 1);

    RCTreeNode *atom = f->atom(j);
    if (atom->parent()->count() > 1)
    {
      if (atom->count() == 1) {
        atom->s_i(VAL_IDX, atom_index);
        atom->s_v(VAL_Z,
                  b_.at(IndexObject::GetSubindex(atom_index)));
        atom->s_v(VAL_F,
                  dList.at(IndexObject::GetSubindex(atom_index)));
      } else
      if (atom->count() > 0)
      {
        atom->s_i(VAL_IDX, atom_index);
      } else { // == 0
        if ((j+1) < f->count())
        {
          atom->s_v(VAL_Z,
                    b_.at(IndexObject::GetSubindex(atom_index)));
          atom->s_v(VAL_F,
                    dList.at(IndexObject::GetSubindex(atom_index)));
        }

        atom->s_i(VAL_IDX, atom_index);
      }
    } else {
      atom->s_i(VAL_IDX, atom->parent()->v(VAL_IDX));
      atom->s_v(VAL_Z, atom->parent()->v(VAL_Z));
      atom->s_v(VAL_F, atom->parent()->v(VAL_F));
    }
  }
}
}

```

```

// задаем p на атомах (так же, как и для случаев 1-5)
haarTree_>root()->s_v(VAL_P, 1.0);
for (int i=1; i <= haarTree_>horizon(); i++) {
    RCTreeFiltration *f = haarTree_>filtration(i);
    for (int j=0; j < f->count(); j++) {
        RCTreeNode *b_atom = f->atom(j);
        if ((j+1) == f->count())
        {
            double sumP = 0.0;
            double q = 0.0;
            for (int k=0; k < f->count() - 1; k++) {
                RCTreeNode *atom = f->atom(k);
                sumP += atom->v(VAL_P);
            }
            q = 1.0 - sumP;
            b_atom->s_v(VAL_P, q);
        } else
        if (b_atom->count() <= 1) {
            b_atom->s_v(VAL_P,
                IndexObject::GetMeasure(b_atom->v(VAL_IDX)));
        }
    }
}

// вычисляем Z
for (int i=0; i < haarTree_>horizon(); i++) {
    RCTreeFiltration *f = haarTree_>filtration(i);
    for (int j=0; j < f->count(); j++) {
        RCTreeNode *atom = f->atom(j);
        if (atom->count() == 2) {
            RCTreeNode *child1 = atom->child(0);
            RCTreeNode *child2 = atom->child(1);
            child2->s_v(VAL_Z, (atom->v(VAL_Z) * atom->v(VAL_P)
                - child1->v(VAL_Z) * child1->v(VAL_P))
                / child2->v(VAL_P));
        }
    }
}

IndexObject::Finalize();

```

```

    return true;
}
...
bool ModelObject::calculationStep3()
{
    if (case_type_ == 6) {
        return calculationStep3_case6();
    } else { // случаи 1-5
        ...
    }
    return false;
}
...
// вычисляем компоненты портфеля
void ModelObject::calculationStep4()
{
    {
        RCTreeNode *p_atom = haarTree_>root();
        p_atom->s_v(VAL_BETA, c_);
        p_atom->s_v(VAL_GAMMA, 0.0);
        p_atom->s_v(VAL_CAPITAL, c_);

        for (int n = 1; n <= haarTree_>horizon(); n++) {
            RCTreeFiltration *f = haarTree_>filtration(n);
            RCTreeNode *a_atom = f->atom(n-1);
            RCTreeNode *b_atom = f->atom(n);

            double sumCkCn_Pk = 0.0;
            double sumBkBn_Pk = 0.0;
            double Cavg = c_;
            double A = haarTree_>root()->v(VAL_Z);

            for (int k=0; k < n-1; k++) {
                RCTreeNode *atom = f->atom(k);
                sumCkCn_Pk += (atom->v(VAL_F) - a_atom->v(VAL_F))
                    * atom->v(VAL_P);
                sumBkBn_Pk += (atom->v(VAL_Z) - a_atom->v(VAL_Z))
                    * atom->v(VAL_P);
            }
        }
    }
}

```

```

a_atom->s_v(VAL_GAMMA,
            (a_atom->v(VAL_F) - Cavg + sumCkCn_Pk)
            / (a_atom->v(VAL_Z) - A + sumBkBn_Pk));
a_atom->s_v(VAL_BETA, a_atom->v(VAL_F)
            - a_atom->v(VAL_GAMMA) * a_atom->v(VAL_Z));
a_atom->s_v(VAL_CAPITAL, a_atom->v(VAL_BETA)
            + a_atom->v(VAL_GAMMA) * a_atom->v(VAL_Z));

b_atom->s_v(VAL_GAMMA, a_atom->v(VAL_GAMMA));
b_atom->s_v(VAL_BETA, a_atom->v(VAL_BETA));
b_atom->s_v(VAL_CAPITAL, b_atom->v(VAL_BETA)
            + b_atom->v(VAL_GAMMA) * b_atom->v(VAL_Z));

RCTreeNode *n_atom = a_atom;
while (n_atom->count() == 1) {
    n_atom->child(0)->s_v(VAL_GAMMA, 0.0);
    n_atom->child(0)->s_v(VAL_BETA, n_atom->v(VAL_CAPITAL));
    n_atom->child(0)->s_v(VAL_CAPITAL, n_atom->v(VAL_CAPITAL));
    n_atom = n_atom->child(0);
}
}

RCSchemeDisplayPropertyGroup *grp =
    haarScheme_>displayPropertyGroup();
grp->addProperty(VAL_BETA, SYMB_BETA, SYMB_BETA);
grp->addProperty(VAL_GAMMA, SYMB_GAMMA, SYMB_GAMMA);
grp->addProperty(VAL_CAPITAL, trUtf8("Капитал"), trUtf8("Капитал"));
haarScheme_>updateSchemeFromTree();
grp->update();

RCMessage(trUtf8("Компоненты_портфеля_вычислены"));
}
...

```

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2016611994

**Построение хеджирующих стратегий посредством
хааровских интерполяций**

Правообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Ростовский государственный строительный университет» (РГСУ) (RU)*

Автор: *Цветкова Инна Владимировна (RU)*



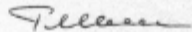
Заявка № 2015662543

Дата поступления 17 декабря 2015 г.

Дата государственной регистрации

в Реестре программы для ЭВМ 16 февраля 2016 г.

Руководитель Федеральной службы
по интеллектуальной собственности

 Г.П. Наливаев