

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»
Кафедра статистического моделирования

Миллер Анастасия Александровна

ИМИТАЦИОННАЯ МОДЕЛЬ АМЕРИКАНСКОГО ОПЦИОНА

Отчет о научно-исследовательской практике

Научный руководитель:

д. ф.-м. н., профессор С. М. Ермаков

Санкт-Петербург

2015

Оглавление

Введение	3
Глава 1. Метод случайных деревьев	4
1.1. Обозначения и умолчания	4
1.2. Построение дерева	5
Глава 2. Устранение экспоненциальной сложности	7
2.1. Анализ распределения состояний с помощью эмпирической функции рас- пределения	7
2.1.1. Описание метода	7
2.1.2. Численные результаты	8
2.2. Конечная сетка состояний	9
2.2.1. Описание метода	9
2.2.2. Численные результаты	10
Список литературы	12
Приложение А. Реализация на Java	13

Введение

Задачами семестра являлись реализация и сравнение методов оценки цены американского опциона. В главе 1 содержится краткое описание исходного метода и оценок, численные результаты реализации, в главе 2 — описание модифицированных методов и оценок, численные результаты их реализации. Глава 2 является описанием результатов работы в течение семестра.

Глава 1

Метод случайных деревьев

1.1. Обозначения и умолчания

Будем строить модель на примере американского опциона с конечным числом дат погашения t_1, \dots, t_m (также называемого бермудским опционом). Мы также сузим класс решаемых нами задач до тех, в которых вся необходимая информация об активе, на который выписан рассматриваемый опцион, может быть представлена в виде Марковского процесса $S(t), t \in \{t_i\}_{i=1}^m$ со значениями в \mathbb{R}^d . Для уменьшения объёма текста будем обозначать $S(t_i) \equiv S_i$. Положим также $h_i(x)$ — дисконтированный размер выплаты по опциону в момент t_i при том, что $x = S_i$ и опцион не был исполнен до этого, $V_i(x)$ — стоимость опциона в момент t_i при том, что $x = S_i$.

Нетрудно видеть, что

$$V_m(x) = h_m(x), \quad (1.1)$$

$$V_{i-1}(x) = \max \{h_{i-1}(x), \mathbb{E}[V_i(S_i) | S_{i-1} = x]\} \quad (1.2)$$

— на каждом шаге мы выбираем наиболее выгодное решение. $V_0(S_0)$ — цена опциона с m датами исполнения.

Оценки В [1] предложены оценки сверху и снизу \hat{V}_0 и \hat{v}_0 и доказана их состоятельность и асимптотическая несмещённость для $V_0(S_0)$.

$$\hat{V}_m^{j_1 \dots j_m} = h_m(S_m^{j_1 \dots j_m}), \quad (1.3)$$

$$\hat{V}_i^{j_1 \dots j_i} = \max \left\{ h_i(S_i^{j_1 \dots j_i}), \frac{1}{b} \sum_{j=1}^b \hat{V}_{i+1}^{j_1 \dots j_i j} \right\}, \quad (1.4)$$

$$\hat{v}_m^{j_1 j_2 \dots j_m} = h(S_m^{j_1 j_2 \dots j_m}), \quad (1.5)$$

$$\hat{v}_{ik}^{j_1 j_2 \dots j_i} = \begin{cases} h(S_i^{j_1 j_2 \dots j_i}), & \text{если } \frac{1}{b-1} \sum_{j=1, j \neq k}^b \hat{v}_{i+1}^{j_1 j_2 \dots j_i j} \leq h(S_i^{j_1 j_2 \dots j_i}), \\ \hat{v}_{i+1}^{j_1 j_2 \dots j_i k}, & \text{иначе} \end{cases} \quad (1.6)$$

$$\hat{v}_i^{j_1 j_2 \dots j_i} = \frac{1}{b} \sum_{k=1}^b \hat{v}_{ik}^{j_1 j_2 \dots j_i}. \quad (1.7)$$

1.2. Построение дерева

Метод случайного дерева основан на моделировании цепи S_0, S_1, \dots, S_n состояний актива. Зафиксируем параметр ветвления b . Из исходного состояния S_0 смоделируем b независимых следующих состояний $S_1^1, S_1^2, \dots, S_1^b$, все с условием S_0 . Для каждого S_1^i снова смоделируем b независимых последующих состояний $S_2^{i1}, \dots, S_2^{ib}$. На m -ом шаге будем иметь b^m состояний, и это и есть источник основного недостатка этого метода — его экспоненциальной алгоритмической сложности. Пример дерева можно увидеть на рис. 1.1.

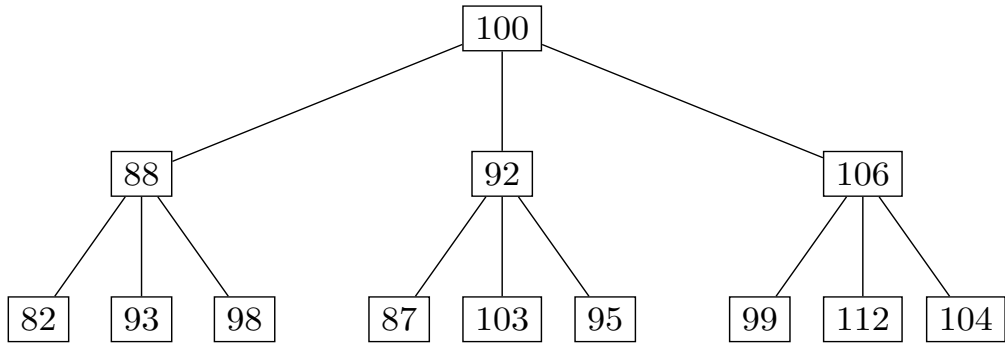


Рис. 1.1. Выбор вершин по квантилям эмпирического распределения

Таким образом, мы имеем дерево с $\sum_{k=1}^m b^k = \frac{b(b^m-1)}{b-1} = O(b^m)$ вершинами и две оценки искомой величины по этому дереву. Классический алгоритм реализован, результаты его работы представлены на рис. 1.2.

Наибольшая частота исполнения опциона, которую классическая оценка способна посчитать при разумных требованиях к памяти — ежемесячно, $n = 12$. Поэтому оценить этим алгоритмом стоимость американского опциона не представляется возможным.

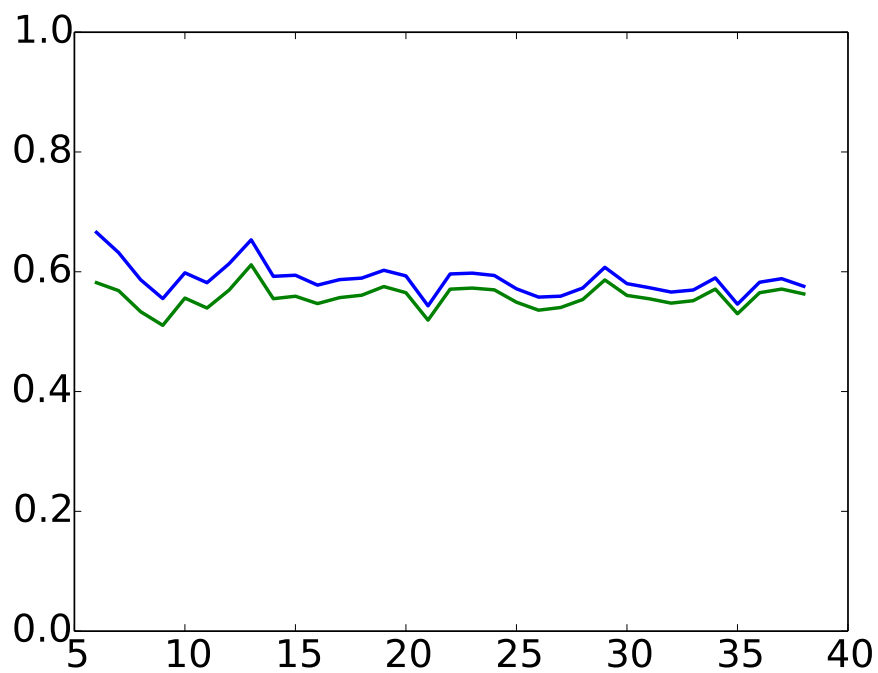


Рис. 1.2. Верхняя и нижняя оценки стоимости опциона по алгоритму Броуди-Глассермана. Оценки стоимости опциона с начальной ценой 100, выписанного на срок 1 год на базовый актив с риск-нейтральной процентной ставкой $r = 0.05$, дивидендной ставкой $\delta = 0.1$ и волатильностью $\sigma = 0.2$, цена которого — случайный процесс, являющийся геометрическим броуновским движением с параметрами $\mu = r - \delta$ и σ , исполняемого 4 раза в году

Глава 2

Устранение экспоненциальной сложности

Начиная с некоторого момента t_k , когда общее число состояний достигнет некоторого n , мы перестанем генерировать дочерние вершины ко всем состояниям. В следующий момент времени, t_{k+1} , мы будем иметь всё так же n состояний, а не bn . Есть несколько способов сделать это, в том числе — ограничивать множество состояний, в которые может перейти базовый актив.

2.1. Анализ распределения состояний с помощью эмпирической функции распределения

2.1.1. Описание метода

В том случае, когда состояние актива S является числом в \mathbb{R}^1 , в качестве параметра X , по распределению которого мы составляем эмпирическую функцию распределения, можно использовать само S , иначе можно использовать $h(S)$.

Пусть мы промоделировали дерево состояний базового актива до момента t_{k-1} . Тогда определено множество состояний актива в момент времени t_{k-1} : $\{S_j\}_{j=1}^n, n = b^{i-1}$. Промоделировав у каждой $j \in 1 : n$ вершины b дочерних вершин (независимых реализаций процесса изменения состояния актива) $\{S_j^i\}_{i=1}^b$, получим множество $\left\{ \{S_j^i\}_{i=1}^b \right\}_{j=1}^n$. Эмпирическая функция распределения состояния базового актива выглядит как

$$F_S(x) = \frac{1}{bk} \# \left\{ (i, j) \in 1 : b \times 1 : k \mid S_i^j < x \right\}.$$

Тогда мы можем сгруппировать вершины по квантилям их эмпирического распределения:

$$\forall k \in 1 : n \quad A_k = \left\{ S_i^j \mid \frac{k-1}{n} \leq F_S(S_i^j) < \frac{k}{n} \right\}.$$

У каждой группы однозначно определена медиана: либо среднее наблюдение в группе, либо смесь двух наиболее близких к середине. Заменяя всех членов группы её медианой, мы получаем bn вершин вместо n . Процесс проиллюстрирован на рис. 2.1.

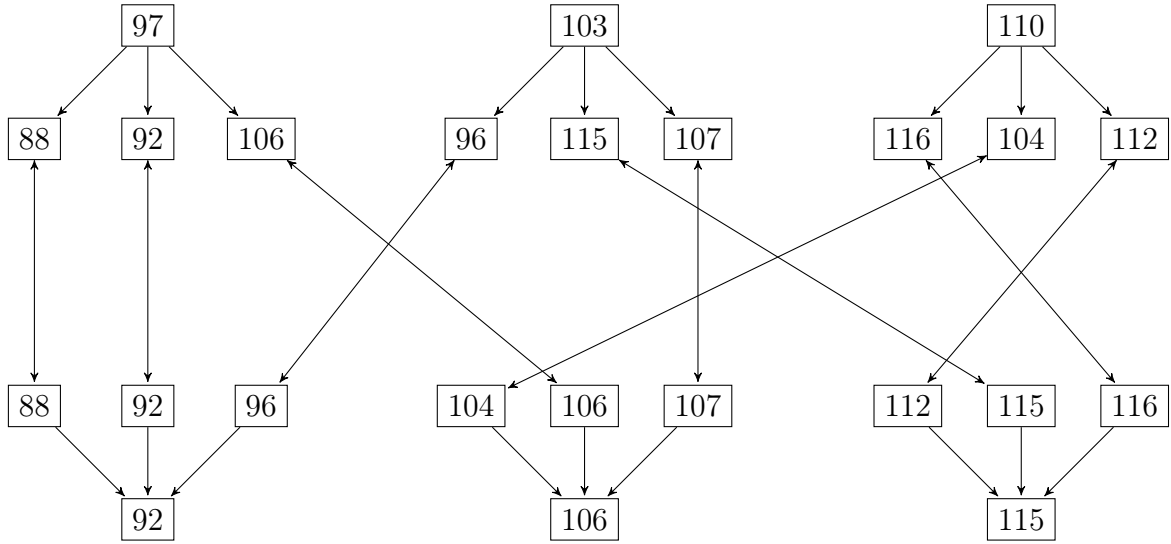


Рис. 2.1. Выбор вершин по квантилям эмпирического распределения

Таким образом, количество рассматриваемых состояний не увеличится. С другой стороны, этот метод предполагает хранение в памяти всего дерева, а не только непосредственно обчитываемой ветки, как это предполагалось в исходной работе [1].

Также стоит отметить, что, пользуясь таким (и подобными ему) методом выбора состояний процесса, мы существенно нарушаем предположение об условной независимости реализаций: теперь состояние актива в t_i зависит не только от состояния актива в t_{i-1} , но и от состояния этого актива в других реализациях процесса. Доказательство сходимости оценок $\hat{V}_0(S_0)$ и $\hat{v}_0(S_0)$ к $V_0(S_0)$ строится на том, что $\hat{V}_{i+1}^{j_1 \dots j_i 1}, \dots, \hat{V}_{i+1}^{j_1 \dots j_i b}$ при данном $S_i^{j_1 \dots j_1}$ — независимые одинаково распределённые случайные величины с математическим ожиданием $E[V_{i+1}(S_{i+1}) | S_i^{j_1 \dots j_1}]$. Следовательно, всё, чего мы можем ожидать, не меняя структуру оценок — достаточно слабой корреляции.

Генерирование каждого следующего ряда занимает $O(bn \log(bn))$ времени (самая затратная операция — сортировка массива сгенерированных вершин длины bn) и $O(bn)$ памяти, следовательно, сложность моделирования в целом составляет $O(mbn \log(bn))$

2.1.2. Численные результаты

Алгоритм также реализован, принципиальная часть представлена в приложении A.1. Результаты моделирования для оценки стоимости того же опциона, что был использован для демонстрации классического алгоритма, представлены на рис. 2.2 Из графика видно, что сходимости к истинному значению не наблюдается.

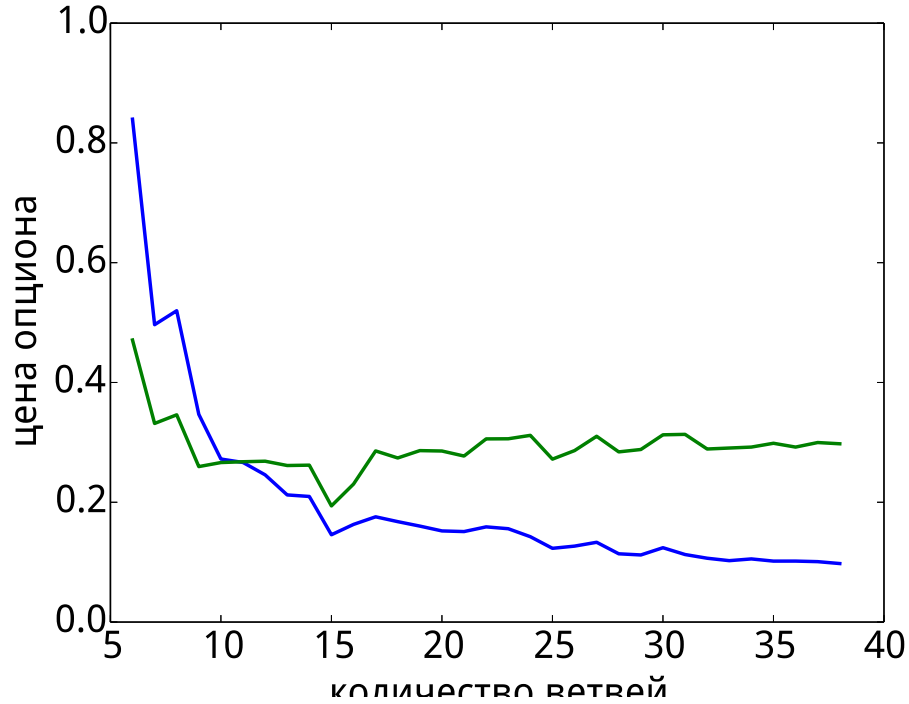


Рис. 2.2. Верхняя и нижняя оценки стоимости опциона по дереву с анализом эмпирической функцией распределения

Оценки стоимости опциона с начальной ценой 100, выпущенного на срок 1 год на базовый актив с риск-нейтральной процентной ставкой $r = 0.05$, дивидендной ставкой $\delta = 0.1$ и волатильностью $\sigma = 0.2$, цена которого — случайный процесс, являющийся геометрическим броуновским движением с параметрами $\mu = r - \delta$ и σ , исполняемого 4 раза в году

2.2. Конечная сетка состояний

2.2.1. Описание метода

Для того, чтобы сложность алгоритма по памяти составляла $O(bm)$, необходимо начиная с некоторого момента ограничивать множество состояний, в которые может перейти актив из данного. Так как мы имеем дело с нестационарным случайным процессом, распределение состояний на следующем шаге меняется, как только мы получаем новую реализацию состояния на предыдущем шаге.

Можно использовать знания о законе распределения, которому подчиняется состояние базового актива, чтобы уменьшить число возможных состояний.

$$dS_t = \mu S_t dt + \sigma S_t dW_t \implies \log\left(\frac{S_t}{S_0}\right) \sim N\left(\left(\mu - \frac{\sigma^2}{2}\right)t, \sigma^2 t\right).$$

Ограничив множество состояний базового актива в момент времени t числом n , мы

можем рассчитать, попадание в какие n классов состояний будет равновероятно. Пусть $F(x)$ — функция распределения $N(0, 1)$, тогда

$$\forall i \in 1 : n \quad \exists \xi_i = F^{-1} \left(\frac{i - 0.5}{n} \right) \text{ — представитель } i\text{-го состояния,} \quad (2.1)$$

$$\forall i \in 0 : n \quad \exists z_i = F^{-1} \left(\frac{i}{n} \right) \text{ — граница } i\text{-го состояния.} \quad (2.2)$$

Для каждого момента времени t_i определено множество состояний

$$S_i^j = S_0 \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma \sqrt{t} \xi_j \right), j \in 1 : n$$

и множество границ этих состояний

$$Z_i^j = S_0 \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma \sqrt{t} z_j \right), j \in 1 : n.$$

Для любого состояния $S_i^{j_1 \dots j_i}$ найдётся $k \in 1 : n : Z_i^{k-1} \leq S_i^{j_1 \dots j_i} < Z_i^k$, тогда $S_i^{j_1 \dots j_i} := S_i^j$. Иллюстрация процесса — на рис. 2.3. Множество состояний не зависит от имеющихся

Рис. 2.3. Дискретизация по квантилям $GBM(\mu, \sigma)$

результатов моделирования, только от параметров базового актива, следовательно, генерируемые траектории будут достаточно независимы, чтобы подходить под условия сходимости оценок (1.3), (1.5). Количество возможных состояний на каждом шаге можно увеличивать по мере увеличения t_i , что может частично компенсировать растущую дисперсию.

2.2.2. Численные результаты

Результаты моделирования для оценки стоимости того же опциона, что был использован для демонстрации классического алгоритма, представлены на рис. 2.4. Несмотря на выполнение условий сходимости, налагаемых на траектории, вычисления демонстрируют отсутствие сходимости при $b \rightarrow \infty$: при больших значениях b ($b = 60, 80$, не показаны на графике) оценки демонстрируют устойчивое поведение, и среднее значение верхней и нижней оценки по 100 реализаций для каждого b равны 0.48 ± 0.02 и 0.43 ± 0.03 соответственно.

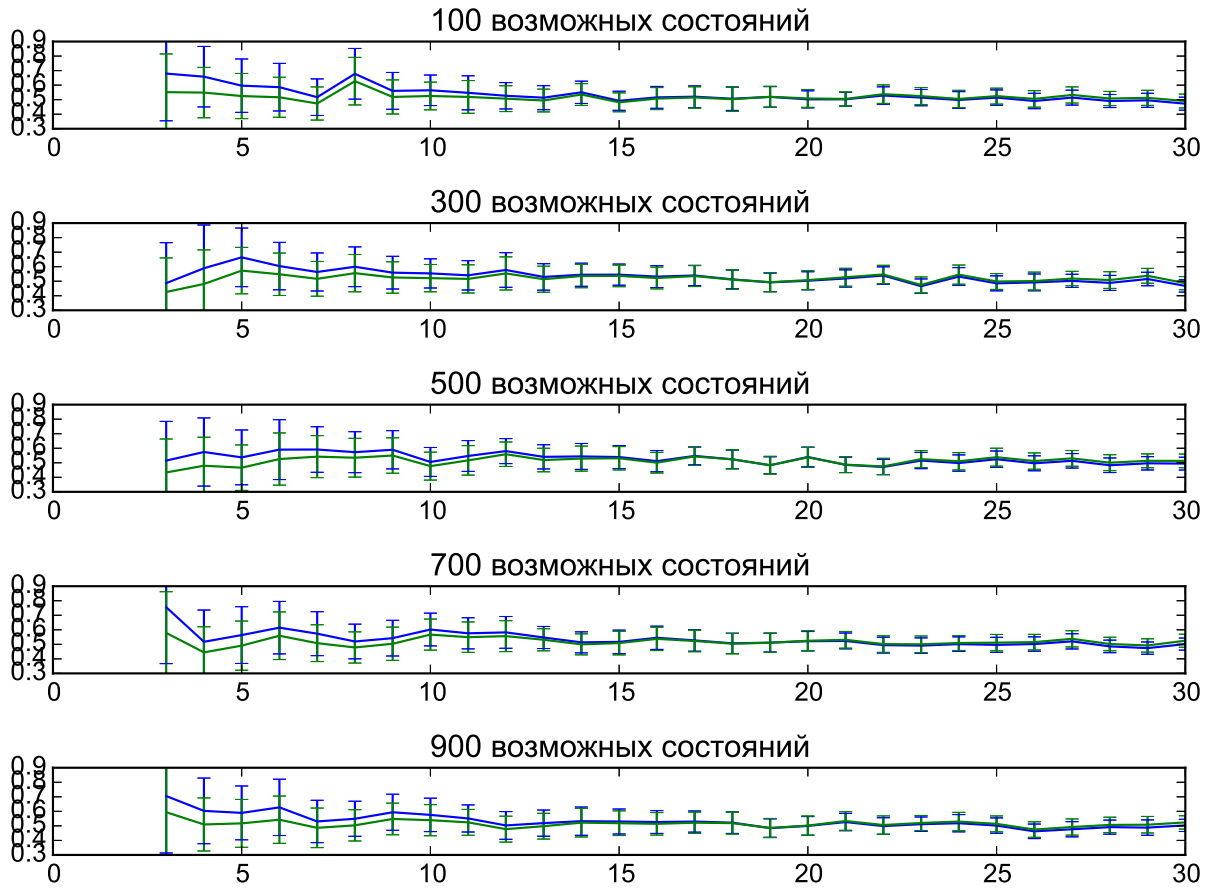


Рис. 2.4. Верхняя и нижняя оценки стоимости опциона по дереву с анализом эмпирической функцией распределения

Оценки стоимости опциона с начальной ценой 100, выписанного на срок 1 год на базовый актив с риск-нейтральной процентной ставкой $r = 0.05$, дивидендной ставкой $\delta = 0.1$ и волатильностью $\sigma = 0.2$, цена которого — случайный процесс, являющийся геометрическим броуновским движением с параметрами $\mu = r - \delta$ и σ , исполняемого 4 раза в году

Дальнейшие планы

1. Посчитать конечную сетку
2. Рассмотреть метод взвешивания траекторий (количество дочерних вершин, которое позволено иметь вершины, пропорционально её вкладу в общую оценку)

Список литературы

1. Broadie M., Glasserman P. Pricing American-style securities by simulation // Journal of Economic Dynamics and Control. — 1997. — Vol. 21. — P. 1323–1352.
2. Broadie Mark, Glasserman Paul, Jain Gautam. Enhanced Monte Carlo estimates for american option prices // Journal of Derivatives. — 1997. — Vol. 5, no. 1 (Fall). — P. 25–44.
3. Duffie Darrell. Dynamic Asset Pricing Theory. — Third Edition edition. — Princeton University Press, 2001.
4. Glasserman Paul. Monte Carlo Methods in Financial Engineering. — Springer, 2004.
5. Haug Espen Gaarder. The Complete Guide to Option Pricing Formulas. — McGraw-Hill Education, 2007. — ISBN: 9780071389976. — URL: <https://books.google.ru/books?id=pSKLNAEACAAJ>.
6. Peskir Goran, Shiryaev Albert. [Optimal Stopping and Free-Boundary Problems](#). Lectures in Mathematics. ETH Zürich. — 1 edition. — Birkhäuser Basel, 2006.
7. Ермаков Сергей Михайлович. Метод Монте-Карло и смежные вопросы. — Наука, 1975.
8. Люю Ю Ду. Методы и алгоритмы финансовой математики / Под ред. Е.В. Чепурин. — БИНОМ. Лаборатория знаний, 2007.
9. Положение о видах производных финансовых инструментов. — 2010. — 04.

Приложение А

Реализация на Java

Моделирование организовано итеративно (а не рекурсивно, как это было бы уместно в исходном методе), потому что необходимая информация о дереве в процессе моделирования — это описание текущего ряда целиком. Поэтому проще всего — пройти вниз по дереву один раз.

Listing A.1. Генерирование дерева состояний актива, на который выписан опцион

```
public static ImitatedAsset generateAssetTree(int branches, int steps, double
initialPrice){
    timedelta = 1. / steps;
    ImitatedAsset ans = new ImitatedAsset(initialPrice, branches, false,
        timedelta);
    ImitatedAsset[] prevRow = getFirstRow(ans, branches);
    for (int step = 0; step < steps; step++) {
        boolean median = branches % 2 == 1;
        ImitatedAsset[] curRow = new ImitatedAsset[branches *
            prevRow.length];
        ImitatedAsset[] newRow = new ImitatedAsset[prevRow.length];
        for (int i = 0; i < prevRow.length; i++) {
            for (int b = 0; b < branches; b++) {
                curRow[i * branches + b] = new
                    ImitatedAsset(getRandomPrice(prevRow[i]), branches,
                        false, timedelta);
                curRow[i * branches + b].parent = prevRow[i];}}
        Arrays.sort(curRow);
        for (int i = 0; i < prevRow.length; i++) {
            ImitatedAsset v;
            if (median) {
                v = curRow[i * branches + branches / 2 + 1];
            } else {
                double price = (
                    curRow[i * branches + branches / 2].price +
                    curRow[i * branches + branches / 2 + 1].price
                ) / 2;
```

```
        v = new ImitatedAsset(price, branches, false, timedelta);}
    for (int b = 0; b < branches; b++) {
        curRow[i * branches + b].parent.addChild(v);
        curRow[i * branches + b] = null;}
    newRow[i] = v;}
    prevRow = newRow;}
return ans;}
```