

Санкт-Петербургский государственный университет  
Прикладная математика и информатика  
Исследование операций и принятие решений в задачах  
оптимизации, управления и экономики

Ульмашев Назар Салаватович

Использование случайных графов в финансовой сфере

Бакалаврская работа

Научный руководитель:  
к. ф. -м. н., ст. пр. Абрамовская Т. В.

Рецензент:  
к. ф. -м. н., доцент Бухвалова В. В.

Санкт-Петербург  
2016

SAINT PETERSBURG STATE UNIVERSITY  
Applied Mathematics and Computer Science  
Operation Research and Decision Making in Optimisation,  
Control and Economics Problems

Ulmashev Nazar

Usage of random graphs in financial analysis

Bachelor's Thesis

Scientific supervisor:  
Senior Lecturer Tatyana Abramovskaia

Reviewer:  
Associate Professor Vera Buhvalova

Saint Petersburg  
2016

# Содержание

<b>1. Введение</b>	<b>4</b>
<b>2. Модель случайного графа с фиксированным распределением степеней вершин</b>	<b>7</b>
2.1. Производящие функции . . . . .	7
2.2. Величина компонент связности . . . . .	11
2.3. Ориентированные графы . . . . .	13
<b>3. Модель кросс-холдинга Гая-Кападия</b>	<b>18</b>
3.1. Общая структура . . . . .	18
3.2. Производящие функции . . . . .	21
3.3. Фазовый переход . . . . .	23
<b>4. Алгоритм моделирования случайного графа и запуск дефолтов</b>	<b>26</b>
4.1. Общая схема . . . . .	26
4.2. Моделирование дискретной случайной величины . . . . .	27
4.3. Преобразование набора вершин в граф . . . . .	29
4.4. Моделирование кросс-холдинга . . . . .	30
4.5. Моделирование дефолтов на полученном графе . . . . .	31
4.6. Программа . . . . .	31
<b>5. Результаты</b>	<b>33</b>

# 1. Введение

Существуют системы, в которых элементы тесно и сложно связаны между собой. Примером такой системы может являться замкнутая группа людей, в которой один участник заразился некоторой инфекцией. Тогда распространение инфекции в группе зависит от частоты контактов заболевшего с остальными членами группы, силы иммунитета и прочих факторов. Аналогичным примером может служить сеть связанных компьютеров, в которой распространяется вирус, или социальные сети и распространение через них различного вирусного контента. Подобные эффекты можно увидеть и в финансовой сфере. Объектом исследования данной работы являются кросс-холдинги. Кросс-холдинг — это система, состоящая из различных организаций (например, банков), в которой участники обладают друг перед другом рядом обязательств (взаимные кредиты в случае банков). Аналогом заболевания в таком случае будет служить дефолт банка, т.е. состояние неплатежеспособности и невозможность отдавать долги. Таким образом, дефолты могут распространяться от банка к банку, а неплатежеспособность единственного банка может привести к краху всей системы. Подробно системы кросс-холдингов были описаны в [1].

В качестве инструмента, который используется для исследования кросс-холдингов, были выбраны случайные графы с фиксированным распределением степеней вершин. Дело в том, что графы сами по себе удобны для представления и изучения различных систем, представимых в виде сетей, а теория случайных графов позволяет изучать некоторые типичные свойства подобных систем. Конкретная модель случайного графа — с фиксированным распределением сте-

пений вершин — также была выбрана неслучайно. Несмотря на то, что, пожалуй, самой известной моделью является модель случайного графа Эрдёш-Реньи (описана в [2]), она мало подходит для исследования кросс-холдингов из-за Пуассоновского распределения степеней вершин, которое нехарактерно для изучаемых моделей. Поэтому, как утверждается в [4], для подобных систем следует использовать схему, которая позволяет строить граф, основываясь на желаемом распределении степеней его вершин. Для этих целей была выбрана модель, предложенная Ньюменом, Строгатцем и Ватсом (мы будем ссылаться на неё по имени первого автора).

Наложение финансовой составляющей на модель случайного графа Ньюмена приводит нас к модели кросс-холдингов, которая описана в статье Гая-Кападии [3]. В ней аппарат производящих функций объединяется с понятием уязвимости банка, и проводятся исследования соответствующих свойств полученных систем.

С практической точки зрения интересен алгоритм программного моделирования случайного графа с заданным законом распределения. Для его реализации требуется решить несколько проблем: моделирование случайных величин, процесс соединения вершин графа, отсеивание плохих случаев. Общая схема была взята из статьи [4]. Задача моделирования дискретной случайной величины была решена классическим методом Уолкера (описан в [5]).

Работа состоит из 5 параграфов. Во втором описываются различные модели случайных графов и аппарат производящих функций, который позволяет их исследовать. В третьем параграфе говорится о модели кросс-холдинга и о применении в ней теории случайных графов. Некоторые теоретические результаты, полученные в этих

двух параграфах, в дальнейшем сравниваются с данными, полученными в результате моделирования. В четвертом параграфе идёт речь о непосредственно моделировании случайных графов и экономических процессов на них. В заключительном, пятом, параграфе анализируются результаты проведенного моделирования.

## 2. Модель случайного графа с фиксированным распределением степеней вершин

Основная проблема модели случайного графа Эрдёш-Реньи заключается в том, что она обладает Пуассоновским распределением степеней вершин. Однако же в моделях, возникающих в реальной жизни (таких как модели кросс-холдингов или модели распространения эпидемий), это правило обычно не выполняется. Как следствие, необходимо выбрать такую модель случайного графа, в которой распределение степеней вершин было бы необходимым.

### 2.1. Производящие функции

Подход, который используется для построения модели случайного графа, будет основан на производящих функциях.

Напомним, что производящие функции — это один из способов задания вероятностного распределения дискретной случайной величины. Удобство данного способа заключается в относительной простоте вычисления моментов случайной величины.

В случае неориентированного графа, производящая функция может быть задана следующим образом:

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k, \quad (1)$$

где  $p_k$  — вероятность того, что случайно выбранная в графе вершина будет иметь степень  $k$ . Причем распределение нормализованно,

поэтому

$$G_0(1) = 1. \quad (2)$$

Зная производящую функцию  $G_0(x)$ , можно посчитать вероятность  $p_k$ :

$$p_k = \frac{1}{k!} \left. \frac{d^k G_0}{dx^k} \right|_{x=0}. \quad (3)$$

Как было сказано выше, производящие функции удобны при вычислении моментов. Например, нетрудно вычислить математическое ожидание степени вершины в графе:

$$z = \sum_k k p_k = G'_0(1). \quad (4)$$

### Пример.

Возьмем граф с 4 вершинами и следующее распределение степеней его вершин:

X	1	2	3
P(X)	0.2	0.6	0.2

Кроме того, наложим ограничение на существование петель и кратных рёбер, что приведет к условию  $P(X = i) = 0, i > 3$ .

Производящая функция тогда будет выглядеть следующим образом:

$$G_0(x) = 0.2x + 0.6x^2 + 0.2x^3. \quad (5)$$



Математическое ожидание степени вершины:

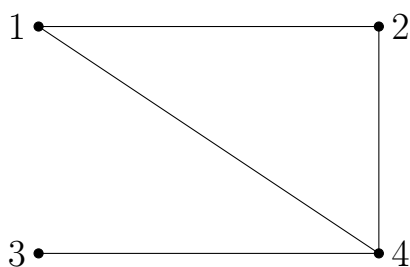
$$z = G'_0(1) = (0.2 + 1.2x + 0.6x^2)|_{x=1} = 1.8. \quad (6)$$

Чтобы получить конкретный граф, удовлетворяющий заданному распределению, нужно сначала посчитать значения случайных величин  $\xi_i, i \in 1 \dots 4$ , соответствующих степени каждой из четырёх вершин.

Пусть, например, выпали следующие значения:

$i$	1	2	3	4
$\xi_i$	2	2	1	3

Тогда, соединив некоторым образом вершины в соответствии с полученными степенями (конкретные способы будут указаны позже), можно получить, например, следующий граф:



Средняя степень вершины в данном случае равна 2, что почти совпадает с полученным мат. ожиданием  $z = 1.8$ .

Если  $G_0$  задает степень случайно выбранной в графе вершины, то распределение ее непосредственных соседей не совпадает с распределением степеней вершин в графе в целом. Для этого рассмотрим распределение степеней вершин, которые можно достигнуть пройдя

по случайно выбранному в графе ребру. Такое ребро входит в некоторую вершину с вероятностью, пропорциональной степени этой вершины, а значит, распределение её степени будет пропорционально величине  $kp_k$ . Нормализованная производящая функция распределения степеней вершин тогда примет следующий вид:

$$G_\alpha(x) = \frac{\sum_k kp_k x^k}{\sum_k kp_k} = x \frac{G'_0(x)}{G'_0(1)}. \quad (7)$$

Если случайным образом выбрать в графе вершину и пройти до  $k$  её ближайших соседей, то вероятности степеней всех таких вершин, без учета ребра, ведущего обратно в исходную вершину, будут распределены в соответствии с функцией  $G_\alpha(x)$ , деленной на  $x$  — последнее для того, чтобы исключить из расчета ребро, по которому мы пришли в такую вершину. Таким образом, пониженная на единицу степень вершины, в которую можно попасть двигаясь по случайно выбранному в графе ребру, распределена в соответствии с функцией:

$$G_1(x) = \frac{G'_0(x)}{G'_0(1)} = \frac{1}{z} G'_0(x). \quad (8)$$

Теперь можно получить производящую функцию для распределения количества непосредственных соседей вершин, смежных с исходной (назовём их смежными вершинами *второго уровня*).

$$\sum_k p_k [G_1(x)]^k = G_0(G_1(x)). \quad (9)$$

Откуда можно получить и математическое ожидание смежных вер-

шин второго уровня:

$$z_2 = \frac{d}{dx} G_0(G_1(x)) \Big|_{x=1} = G'_0(1)G'_1(1) = G''_0(1). \quad (10)$$

Возвращаясь к примеру:

$$z_2 = (1.2 + 1.2x) \Big|_{x=1} = 2.4. \quad (11)$$

Фактическое количество вершин второго уровня в полученном графе (обозначим как  $deg_2$ ):

$i$	1	2	3	4
$deg(i)_2$	3	3	2	2

Таким образом, среднее значение количества вершин второго уровня равняется 2.5.

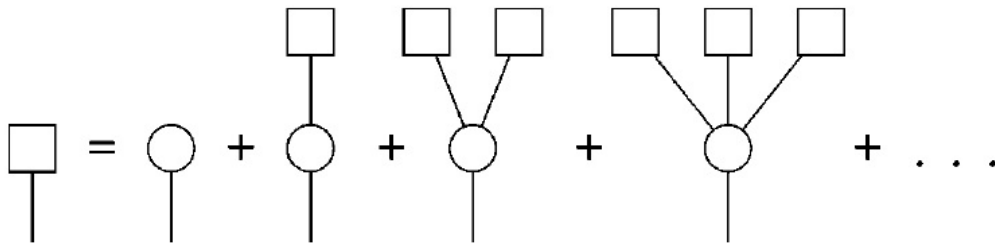
**Замечание.** Несмотря на то, что среднее количество смежных вершин выражается через  $G'_0(1)$ , а среднее количество смежных вершин на втором уровне через  $G''_0(1)$ , предположение о том, что среднее количество смежных вершин уровня  $m$  может быть вычислено как  $G_0^{(m)}(1)$ , не является верным.

## 2.2. Величина компонент связности

Предметом рассмотрения в данном разделе является распределение размеров компонент связности в графе. (Данный параграф представляет исключительно теоретический интерес, поскольку в системах кросс-холдингов предполагается наличие связных графов).

Для начала введем функцию  $H_1(x)$  — производящую функцию для

распределения размеров компонент связности (при этом не рассматривается случай наличия в графе *гигантской компоненты*, т. е. такой наибольшей компоненты, размер которой линейно зависит от размера графа. Граф не имеет гигантской компоненты, если с ростом количества вершин размер компонент связности остается постоянным), которые можно достичь, если двигаться по случайно выбранному ребру в одном из направлений. Учитывая, что вероятность возникновения цикла в отдельной компоненте будет пропорциональна  $N^{-1}$ , при больших значениях  $N$  этой вероятностью можно пренебречь. Таким образом, компоненты связности, порождаемые производящей функцией  $H_1(x)$  может быть представлено в виде рисунка:



Каждая компонента связности имеет древовидную структуру и может состоять либо из единственной вершины  $v \in V$ , в которую можно попасть двигаясь по исходному ребру  $e \in E$ , либо из компонент, в которые можно попасть двигаясь по смежным с  $v$  рёбрам (но не по  $e$  в обратную сторону). Причем эти компоненты будут иметь аналогичное распределение размеров. Теперь обозначим за  $q_k$  вероятность того, что вершина  $v$  имеет  $\deg(v) = k + 1$  (куда входит исходное случайно выбранное ребро и еще  $k$  исходящих из этой вершины рёбер). Тогда верно следующее соотношение:

$$H_1(x) = xq_0 + xq_1H_1(x) + xq_2[H_1(x)]^2 + \dots \quad (12)$$

Причем  $q_k$  — это коэффициент при  $x^k$  для функции  $G_1(x)$ . Таким образом, с учётом вида  $G_1(x)$  в (8):

$$H_1(x) = x \sum_k q_k H_1^k(x) = xG_1(H_1(x)). \quad (13)$$

Если случайным образом выбрать вершину в графе, то на конце каждого исходящего из неё ребра будет подобная компонента связности и, таким образом, порождающая функция для размера всей компоненты имеет следующий вид:

$$H_0(x) = xG_0(H_1(x)). \quad (14)$$

В итоге, зная функции  $G_0(x)$  и  $G_1(x)$ , можно решить уравнения (13) и (14) и найти производящую функцию  $H_0(x)$ .

### 2.3. Ориентированные графы

Объектов нашего дальнейшего изучения будут являться ориентированные графы с фиксированным распределением степеней вершин. Примером такого графа как раз является основной предмет изучения настоящей работы — кросс-холдинг банков.

Ориентированные графы имеют ряд особенностей, которые отличают их от неориентированных графов, и из-за которых требуется несколько модифицировать аппарат, рассмотренный в предыдущих пунктах. Во-первых, в ориентированном графе нет понятия степени вершины в классическом смысле. Вместо этого появляются понятия полустепеней захода и выхода вершины, определяемые соответственно как число входящих и исходящих вершин. Во-вторых, исчезает по-

нятие компоненты связности, поскольку некоторая вершина может быть достигнута из исходной, а обратное не обязано быть правдой. Следует рассматривать отдельно множество всех вершин, достижимых из данной (*компонента выхода*), и множество всех вершин, из которых данная вершина достижима (*компонента захода*). Таким образом, основная идея модификации введенного аппарата — рассматривать отдельно входящие и исходящие ребра.

Пусть  $p_{jk}$  — вероятность того, что случайно выбранная в графе вершина имеет полустепень захода  $j$  и полустепень выхода  $k$ . Причем стоит заметить, что эти степени, вообще говоря, не являются независимыми величинами, т.е.  $p_{jk} \neq p_j p_k$ . Иллюстраций этого свойства может служить Интернет, для которого узел, имеющий большое количество входящих ребёр, как правило имеет и большое количество исходящих.

По аналогии с производящими функциями для неориентированных графов, определим производящую функцию для ориентированного графа:

$$G(x, y) = \sum_{jk} p_{jk} x^j y^k. \quad (15)$$

Поскольку каждое ребро в ориентированном графе выходит из одной вершины и входит в другую, то входы и выходы из вершины в среднем сбалансированы.

$$\sum_{jk} p_{jk} (j - k) = 0. \quad (16)$$

А значит:

$$\left. \frac{\partial G}{\partial x} \right|_{x,y=1} = \left. \frac{\partial G}{\partial y} \right|_{x,y=1} = z. \quad (17)$$

Где  $z$ , как и раньше, — это математическое ожидание полустепени вершины в графе.

### Пример.

Возьмем следующее распределение полустепеней (элементом  $j$ -й строки и  $k$ -го столбца является  $p_{jk}$ ):

	1	2
1	0.5	0.2
2	0.2	0.1

Производящая функция тогда примет следующий вид:

$$G(x, y) = 0.5xy + 0.2x^2y + 0.2xy^2 + 0.1x^2y^2. \quad (18)$$

Аналогично предыдущему примеру возьмем какой-нибудь граф из данного распределения:



И вычислим для него математическое ожидание полустепени:

$$z = \left. \frac{\partial G}{\partial x} \right|_{x,y=1} = 0.5 + 0.4 + 0.2 + 0.2 = 1.3, \quad (19)$$

что очень близко к реальному среднему значению полустепени в данном графе, равному 1.25.

Аналогично предыдущим пунктам, можно определить производящие функции  $G_0$  и  $G_1$ . Только теперь они будут отвечать исключительно за исходящие ребра и связанные с ними характеристики. Для входящих ребер заведем аналогичные производящие функции  $F_0$  и  $F_1$ . Определим их при помощи уже известной функции  $G(x, y)$ :

Производящая функция распределения полустепени захода случайно выбранной вершины:

$$F_0(x) = G(x, 1). \quad (20)$$

Производящая функция для полустепени захода вершины, в которую можно попасть, двигаясь по случайно выбранному в графе ребру:

$$F_1(x) = \frac{1}{z} \left. \frac{\partial G}{\partial x} \right|_{y=1}. \quad (21)$$

Производящая функция распределения полустепени выхода случайно выбранной вершины:

$$G_0(y) = G(1, y). \quad (22)$$

Производящая функция для полустепени выхода вершины, в которую можно попасть, двигаясь по случайно выбранному в графе ребру:



ру:

$$G_1(y) = \frac{1}{z} \frac{\partial G}{\partial x} \Big|_{x=1}. \quad (23)$$

Выражение для среднего числа соседей второго уровня  $z_2$  получим при помощи уравнений (17) и (10):

$$z_2 = G_0''(y) = \frac{\partial^2 G(1, y)}{\partial y^2} \Big|_{x=1} = \frac{\partial^2 G(x, y)}{\partial y^2} \Big|_{x, y=1} = \frac{\partial^2 G(x, y)}{\partial y \partial x} \Big|_{x=1}. \quad (24)$$

Распределение размеров компонент выхода, достижимых из случайно выбранной в графе вершины, задается функцией  $H_0(y)$ :

$$H_0(y) = yG_0(H_1(y)), \quad (25)$$

где  $H_1(y) = y\mathcal{G}_1(H_1(y))$  аналогично случаю неориентированных графов. Аналогичные равенства получаются и для компонент захода.

## 3. Модель кросс-холдинга Гая-Кападия

### 3.1. Общая структура

Систему из банков удобно представлять в виде ориентированного графа. В таком случае вершины графа представляют сами банки, а ребра — межбанковские связи. Входящие рёбра обозначают межбанковские активы (например деньги, которые другие банки должны данному банку), а исходящие — долговые обязательства банка. Таким образом, направление ребра однозначно устанавливает должника и кредитора.

Рассмотрим модель Гая-Кападия на ориентированном графе  $G(V, E)$ ,  $|V| = N$ , где  $N$  достаточно велико. Для каждого  $i \in V$  вводятся следующие величины, отражающие упрощенный балансовый отчёт каждого банка:

- $A_i^{IB}$  — межбанковские активы.
- $A_i^M$  — внешние активы.
- $L_i^{IB}$  — межбанковские пассивы.
- $D_i$  — капитал, находящийся на счетах клиентов.
- $K_i$  — собственные средства банка. Вычисляются по формуле:

$$K_i = A_i^{IB} + A_i^M - L_i^{IB} - D_i. \quad (26)$$

Теперь нужно задать веса  $w(e)$  для каждого  $e \in E$ . В модели Гая-Кападия на межбанковские активы приходится 20 процентов всех

активов банка. Если условно принять сумму всех активов банка за 1, то величина межбанковских активов  $A_i^{IB} = 0.2$ . Причем, для каждого банка они распределены равномерно между должниками, и тогда вес каждого ребра задается следующим образом:

$$\forall e = (v, u) \in E : w(e) = \frac{0.2}{|E_{in}^u|}, \quad (27)$$

где  $E_{in}^u = \{e \in E | e = (x, u), x \in V\}$

Поскольку  $A_i^{IB} + A_i^M = 1$ , то  $A_i^M = 0.8$ . Межбанковские пассивы  $L_i^{IB}$  определяются следующим образом:

$$L_i^{IB} = \sum_{e \in E_{out}^i} w(e), \quad (28)$$

где  $E_{out}^i = \{e \in E | e = (i, x), x \in V\}$

Банк  $i$  будем называть *платежеспособным*, если:

$$(1 - \varphi)A_i^{IB} + qA_i^M - L_i^{IB} - D_i > 0. \quad (29)$$

где  $\varphi$  - доля банков, у которых имеются обязательства к банку  $i$  и которые при этом оказались неплатежеспособными, а  $q$  — цена, по которой можно продать внешние активы.

Неравенство (29) можно переписать в виде:

$$\varphi < \frac{K_i - (1 - q)A_i^M}{A_i^{IB}}, A_i^{IB} \neq 0. \quad (30)$$

Изначально система находится в равновесии, т. е. все банки являются платежеспособными. Если по каким-то причинам неравенство (29)

перестало выполняться, то тогда банк  $i$  будем называть *неплатежеспособным* или, что эквивалентно, *объявившем дефолт*. Причинами выхода системы из состояния покоя могут стать различные внешние факторы: мошенничество (например, как в случае с банком Varigns) или общее снижение внешних активов вкуче с большими потерями для одного из банков. А дефолт одного из участников системы может вызвать целую волну дефолтов, которая будет распространяться по системе.

Обозначим за  $j_i$  полустепень захода для банка  $i$ . Поскольку каждый банк теряет  $1/j_i$  от своих межбанковских активов при дефолте одной из связанных с ним компаний, то из (30) следует, что единственный случай, при котором дефолт может распространиться по системе — это если один из смежных с  $i$  банков оказался неплатежеспособен, а для самого  $i$  верно:

$$\frac{K_i - (1 - q)A_i^M}{A_i^{IB}} < \frac{1}{j_i}. \quad (31)$$

Введем еще одно определение: банк называется *уязвимым*, если он может стать неплатежеспособным при дефолте хотя бы одного из своих соседей. В противном случае банк будет называться *надежным*.

Вероятность того, что банк  $i$  с полустепенью захода  $j$  уязвим:

$$v_j = P\left(\frac{K_i - (1 - q)A_i^M}{A_i^{IB}} < \frac{1}{j}\right), \forall j \geq 1. \quad (32)$$

## 3.2. Производящие функции

Теперь, имея в распоряжении такое понятие как уязвимый банк, составим формальный степенной ряд с коэффициентами, характеризующими вероятность появления уязвимого банка с полустепенями захода и выхода, соответственно,  $j$  и  $k$

$$\mathcal{G}(x, y) = \sum_{jk} v_j p_{jk} x^j y^k. \quad (33)$$

Кроме того, можно определить и производящую функцию  $\mathcal{G}_0(y)$  для распределения полустепеней выхода уязвимого банка, которая будет удовлетворять ранее полученному уравнению (22), но с учетом уязвимости банков примет вид:

$$\mathcal{G}_0(y) = \mathcal{G}(1, y) = \sum_{jk} v_j p_{jk} y^k. \quad (34)$$

Стоит также заметить, что:

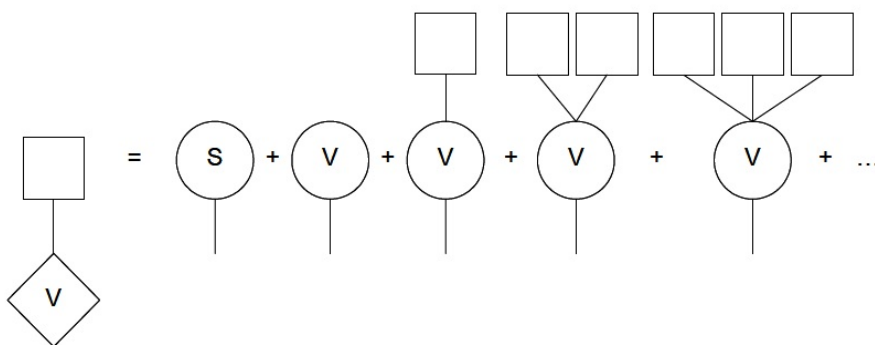
$$\mathcal{G}_0(1) = \mathcal{G}(1, 1) = \sum_{jk} v_j p_{jk} \leq 1. \quad (35)$$

Таким образом,  $\mathcal{G}_0(1)$  дает долю уязвимых банков.

Можно определить и еще одну производящую функцию  $\mathcal{G}_1(y)$  для набора вероятностей полустепеней исхода вершины, которую можно достигнуть, двигаясь по случайно выбранному ребру:

$$\mathcal{G}_1(y) = \sum_{j,k} v_j r_{jk} y^k = \frac{\sum_{j,k} v_j j p_{jk} y^k}{\sum_{j,k} j p_{jk}}. \quad (36)$$

Теперь возьмём некоторый уязвимый банк  $v$  и пройдем по случайно выбранному ребру  $e = (v, u)$ . Если вершина  $u$  является надежным банком, то остановимся, а если уязвимым, то повторим процедуру. Все банки, которые можно достигнуть таким образом из исходной вершины  $v$  назовём *исходящим кластером уязвимости*. Структура такого кластера представлена на рисунке:



Обозначим за  $H_1(y)$  производящую функцию последовательности вероятностей достижения исходящего кластера уязвимости заданного размера (в смысле количества уязвимых банков, которые он содержит) при движении по случайно выбранному исходящему из уязвимо-го банка ребру. Как видно из рисунка, суммарная вероятность будет состоять из вероятности попасть в надежный банк, вероятности попасть в уязвимый банк без исходящих рёбер и т. д. Производящая функция будет выглядеть следующим образом:

$$H_1(y) = \hat{p} + y \sum_{jk} v_j r_{jk} (H_1(y))^k. \quad (37)$$

где  $\hat{p}$  — вероятность попасть в надёжный банк.

Подставив в уравнении (36) функцию  $H_1(y)$  вместо  $y$  и учитывая, что  $\hat{p} = 1 - G_1(1)$ , поскольку  $G_1(1)$  — вероятность перейти в один

уязвимый банк из другого уязвимого банка, получим:

$$H_1(y) = 1 - \mathcal{G}_1(1) + y\mathcal{G}_1(H_1(y)). \quad (38)$$

Осталось описать распределение случайной величины, которая отвечает за размер исходящего кластера уязвимости, которому принадлежит случайно выбранный банк. Ему будет соответствовать производящая функция  $H_0(y)$ . Есть два варианта, которые могут возникнуть при случайном выборе банка в графе: во-первых, выбранный банк может оказаться надёжным. Во-вторых, с вероятностью  $v_j p_{jk}$  он может оказаться уязвимым и иметь полустепени захода и выхода  $j$  и  $k$  соответственно. Во втором случае каждое исходящее ребро ведёт в некоторый исходящий кластер уязвимости, чей размер описывается функцией  $H_1(y)$ . Следовательно, функция  $H_0(y)$  будет выглядеть следующим образом:

$$H_0(y) = \bar{p} + y \sum_{jk} v_j p_{jk} (H_1(y))^k = 1 - \mathcal{G}_0(1) + y\mathcal{G}_0(H_1(y)). \quad (39)$$

где  $\bar{p}$  — вероятность того, что выбранный банк оказался надёжным.

### 3.3. Фазовый переход

Хотя и не всегда можно получить выражение для производящих функций в явной форме, но зато можно посчитать их моменты. В частности, средний размер исходящего кластера уязвимости равен:

$$\mathcal{S} = H'_0(1). \quad (40)$$

Взяв производную в уравнении (39) в точке 1 и учитывая, что производящая функция  $H_1(y)$  нормализована, получаем:

$$H'_0(1) = \mathcal{G}_0[H_1(1)] + \mathcal{G}'_0[H_1(1)]H'_1(1) = \mathcal{G}_0(1) + \mathcal{G}'_0(1)H'_1(1). \quad (41)$$

Аналогично возьмем производную в уравнении (38):

$$H'_1(1) = \frac{\mathcal{G}_1(1)}{1 - \mathcal{G}'_1(1)}. \quad (42)$$

И, подставляя (42) в (41), получаем:

$$\mathcal{S} = \mathcal{G}_0(1) + \frac{\mathcal{G}_1(1)\mathcal{G}'_0(1)}{1 - \mathcal{G}'_1(1)}. \quad (43)$$

Как видно из (46) при  $\mathcal{G}'_1(1) = 1$  происходит *фазовый переход*, т. к. в этой точке присутствует особенность. По-другому это равенство, учитывая (36) и соотношение  $z = \sum_{j,k} jp_{jk}$ , можно записать так:

$$\sum_{j,k} jkv_j p_{jk} = z. \quad (44)$$

где  $z$  — средняя полустепень захода (которая, однако, совпадает со средней полустепенью выхода).

Предположим, что значение  $z$  велико. В этом случае наибольшие значения  $p_{jk}$  будет принимать при больших  $j$  и  $k$ . Таким образом, левая часть уравнения (44) будет увеличиваться посредством  $jkp_{jk}$ , но уменьшаться из-за члена  $v_j$ , поскольку чем больше  $j$ , тем меньше  $v_j$  — это следует из (32). Значит, уравнение (44) будет иметь либо 2 решения, либо ни одного. В первом случае возникают непрерывный отрезок значений  $[z_1, z_2]$ , для которых возможно возникновение



волны дефолтов. Если же значение средней степени лежит вне этого отрезка, то:

- $z < z_1$ : член  $jkp_{jk}$  слишком мал и сеть недостаточно связна для распространения заражения (хорошим примером этого случая является сеть без ребер — в ней распространение дефолтов невозможно).
- $z > z_2$ :  $v_j$  слишком мало, и это означает, что в сети присутствует слишком много безопасных банков, что также не позволит распространяться волне дефолтов.

## 4. Алгоритм моделирования случайного графа и запуск дефолтов

### 4.1. Общая схема

Модель случайного графа Ньюмена предполагает наличие заранее известного закона распределения степеней вершин. Поскольку каждая вершина имеет полустепени захода и выхода, то наша задача - промоделировать  $2N$  значений случайной величины при  $N$  вершинах графа. Случайные величины, отвечающие за эти степени, являются, вообще говоря, независимыми, поэтому достаточно научиться моделировать случайную величину с фиксированным распределением, а потом получить набор из  $2N$  значений этой случайной величины. Однако, из полученного набора не всегда возможно собрать граф. Проверка допустимости осуществляется через равенство:

$$\sum_{i=1}^N j_i = \sum_{i=1}^N k_i, \quad (45)$$

где  $j_i$  — полустепень выхода  $i$ -й вершины, а  $k_i$  — её полустепень захода.

В случае, если данное равенство не выполняется, можно промоделировать набор из  $2N$  значений еще раз. На практике же при больших значениях  $N$  вычисление подходящего набора может занять слишком много времени, поэтому достаточно изменить несколько значений  $u_i$  и  $v_i$  так, чтобы набор оказался допустимым.

## 4.2. Моделирование дискретной случайной величины

В качестве закона распределения степеней вершин для моделирования был выбран степенной закон распределения, который задаётся следующим образом:

$$p_k = Ck^{-\tau}, \quad (46)$$

где  $C$  — коэффициент нормировки, а  $\tau$  — параметр распределения. Конкретные значения  $\tau$  были выбраны данные из статьи [6], которые соответствуют финансовым параметрам в России. Причем для распределений полустепеней входа и захода значения  $\tau$  различны:  $\tau_{in} = 1.92$ , а  $\tau_{out} = 2.64$ . Кроме того, распределение было сдвинуто таким образом, чтобы минимальная полустепень вершины была равна 4.

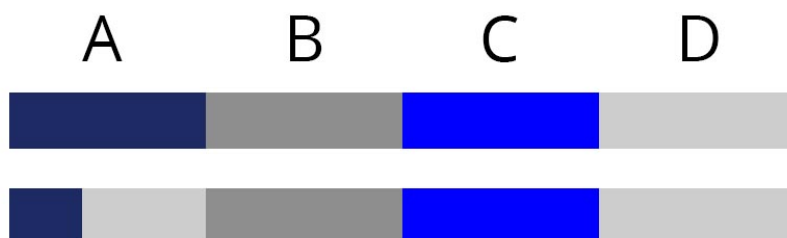
Для моделирования дискретной случайной величины был выбран метод Уолкера, который описан в [5]. Это один из наиболее удобных методов моделирования случайной величины. Основная идея заключается в следующем: пусть имеется некоторое дискретное распределение с  $K$  исходами. Далее берется равномерное распределение с таким же количеством исходов. Те исходы  $i$  в оригинальном распределении, для которых  $P(i) > \frac{1}{K}$  назовём *реципиентами*, а для которых  $P(i) < \frac{1}{K}$  *донорами*. Причем *излишки* доноров равны  $P(i) - \frac{1}{K}$ . Для каждого донора, зная его излишки, можно завести массив порогов, который будет указывать на то, какому реципиенту передавать лишнюю вероятность. Таким образом, один донор может передавать "лишнюю" вероятность сразу нескольким реципиентам. Имея вычис-

ленный массив порогов, необходимо запустить счетчик случайных чисел и узнать значение равномерно распределенной случайной величины. Если её значение попадает в ту часть, которая является излишком для донора, то по массиву порогов вычисляется реципиент, которому нужно передать "лишнюю" вероятность, и считается, что случайная величина приняла значение, соответствующее этому реципиенту.

### Пример.

Пусть некоторое распределение с 4 исходами задается следующими вероятностями:

X	A	B	C	D
P(X)	0.1	0.25	0.25	0.4



Здесь исход  $A$  - донор с излишком в 0.15. Заведем для него массив порогов, который в данном случае будет состоять из единственного значения 0.25. Таким образом, если равномерно распределенная случайная величина попадет на исход  $A$ , но превысит пороговое значение (на рисунке соответствует правой части полосы события  $A$ ), то будет считаться, что на самом деле выпал исход  $D$ .

### 4.3. Преобразование набора вершин в граф

На данном этапе мы располагаем набором из  $2N$  вершин и значениями их полустепеней  $j_i$  и  $k_i$ , которые были взяты из заданного распределения. Теперь требуется построить множество рёбер  $E$  графа таким образом, чтобы полустепени полученного графа  $G(V, E)$   $v_{in}$  и  $v_{out}$  были в точности такими как и смоделированные значения  $j_i$  и  $k_i$  соответственно. Можно рассмотреть несколько вариантов соединения узлов графа:

- На каждой итерации равномерно и случайно выбирать одну из вершин, которой нужно входящее ребро, и таким же образом выбирать вершину, которой нужно выходящее ребро. После этого нужно попытаться их соединить. При этом не всегда будет возможность поставить ребро между выбранной парой, т.к. такое ребро уже может быть в графе (т.е. появляется опасность возникновения кратных рёбер, которые не предполагаются в моделях кросс-холдингов), или ребро может быть потенциальной петлёй. Плюсом данного метода является быстрая скорость выбора вершины —  $O(1)$  в каждом случае, однако общее время метода излишне велико из-за того, что на последних итерациях алгоритм пытается выбрать недопустимые рёбра и приходится выбирать вершины снова.
- Второй вариант — на каждой итерации составлять список допустимых рёбер. В таком случае при постановке каждого ребра приходится проводить предподсчет - проверять все пары вершин за  $O(N^2)$ , однако выбор ребра происходит равновероятно из множества рёбер, которые точно являются допустимыми. Данный метод можно улучшить, если на каждой итерации поль-

зоваться множеством допустимых рёбер из прошлой итерации, пересчитывая его за  $O(N)$  при хранении множества в виде односвязного списка.

#### 4.4. Моделирование кросс-холдинга

Следующий шаг после расстановки рёбер — присвоить каждому  $e \in E$  некоторый вес  $w(e) > 0$ . Для этого достаточно для каждой вершины вычислить количество входящих в неё рёбер. Для всего графа нагрузка рёбер будет произведена за

- $O(N^3)$ , если граф хранится в виде списка смежности.
- $O(N^2)$ , если граф хранится в виде матрицы смежности.
- $O(N|E|)$ , если граф хранится в виде списка рёбер.

Далее необходимо вычислить начальные значения элементов балансового отчета. Значения  $A_i^{IB}$  и  $A_i^M$  задаются для каждой вершины одинаково: по 0.2 и 0.8 соответственно. Для подсчёта размера межбанковских пассивов  $L_i^{IB}$  нужно для каждой вершины подсчитать суммарный вес исходящих рёбер. При известном  $L_i$  немедленно вычисляется и размер депозитов  $D_i$  и, таким образом, для различных структур данных, используемых для хранения графа, время подсчёта будет следующее:

- $O(N)$ , для списка смежности.
- $O(N)$ , для матрицы смежности.
- $O(|E|)$ , для списка рёбер.

## 4.5. Моделирование дефолтов на полученном графе

Получив некоторый граф из заданного распределения, на нём можно запустить дефолт. Для этого выбирается некоторая вершина (или сразу несколько вершин) в графе, и соответствующий банк объявляется неплатежеспособным, а все ребра, инцидентные данной вершине, теряют свой вес и пересчитываются балансовые значения для каждого узла. Далее проверяется условие платежеспособности (29) для всех смежных банков, и если какие-то банки на этой итерации объявили дефолт, то та же процедура выполняется и для них. Данный процесс выполняется итеративно до тех пор, пока либо вся система не рухнет, либо распространение дефолтов не остановится.

## 4.6. Программа

Программа, моделирующая графы и распространение дефолтов на них, написана на языке C++ и состоит из 5 модулей.

- `Dir.cpp` — Основной файл программы.
- `Edge.h` — Заголовочный файл для структуры, объектом которой является ребро графа.
- `walker.h` — Заголовочный файл, в котором происходит моделирование случайной величины.
- `Graph.h` — Заголовочный файл для структуры, объектом которой является граф.

- `Set_of_degrees.h` — Заголовочный файл для структуры, объектом которой является множество степеней вершин.

Код программы доступен по ссылке:

[https://github.com/Ulmashev/random\\_graphs](https://github.com/Ulmashev/random_graphs)



## 5. Результаты

Ниже представлены графики, иллюстрирующие некоторые результаты моделирования. Для значения каждого параметра была взята выборка из 50 графов. В начале рассмотрим влияние изменения параметров распределения непосредственно на структуру графа, а именно на число ребер:

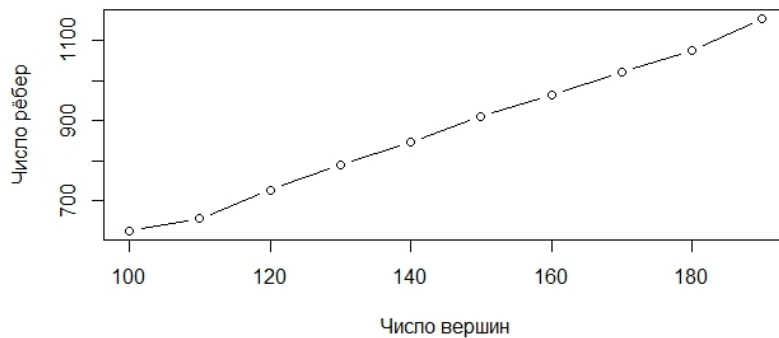


График 1

График 1 иллюстрирует явную линейную зависимость между числом вершин и числом ребер в графе при заданном в (46) распределении.

Далее представлены графики зависимости числа ребер от смещения параметров  $\tau_{in}$ ,  $\tau_{out}$  и максимальной полустепени в графе:

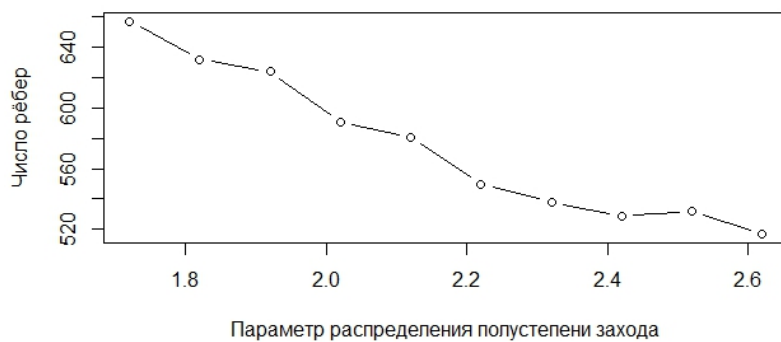


График 2

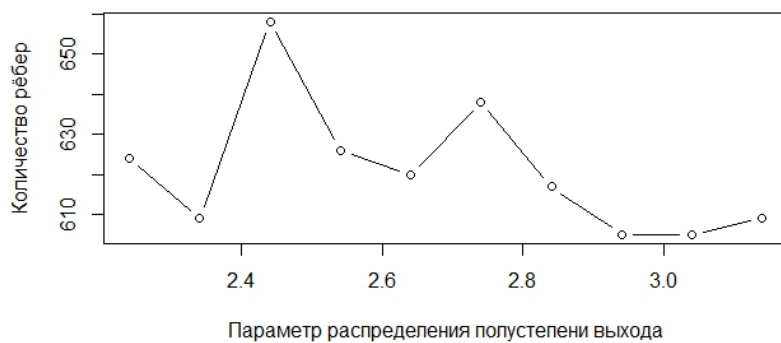


График 3

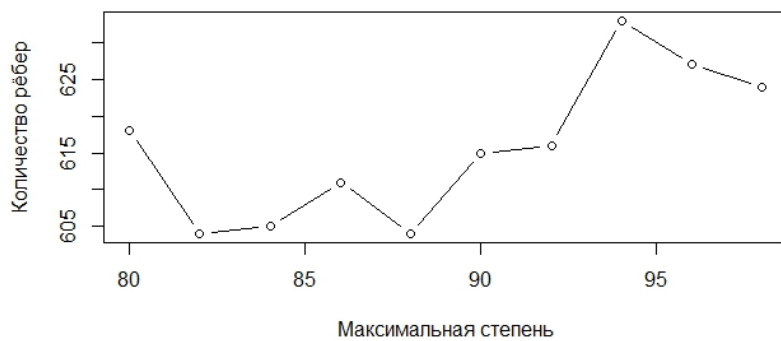


График 4

Далее рассмотрим влияние различных параметров распределения на количество банков, которые стали неплатежеспособными после дефолта случайно выбранного в графе банка. Это количество выражается в процентном соотношении относительно размера всей системы:

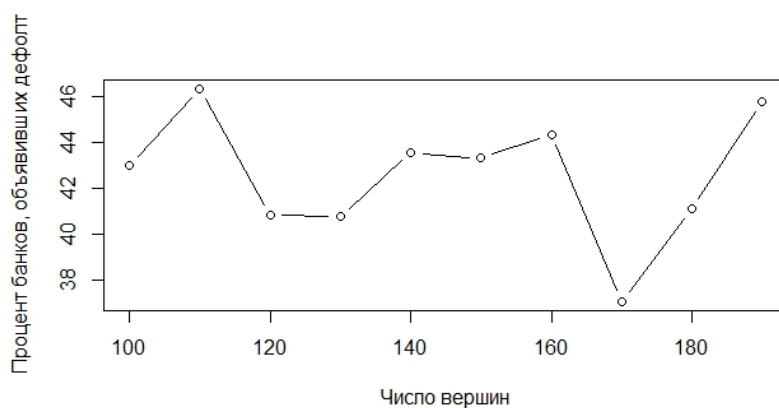


График 5

Как видно из графика 5, процент банков, объявивших дефолт при таких параметрах распределения, примерно одинаков для всех значений  $N$  и находится в районе 42 процентов. Причем, если увеличить минимальное значение полустепени с 4 до 5, то при тех же параметрах моделирования дефолт исходного банка никак не влияет на систему.

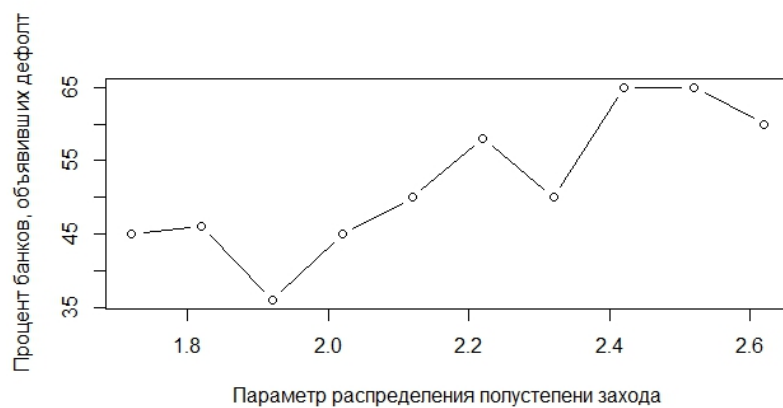


График 6

График 6 иллюстрирует тот факт, что с увеличением  $\tau_{in}$  количество дефолтов становится чрезвычайно большим, что позволяет сделать вывод о желательном понижении этого параметра.



График 7

Небольшие ограничения максимальной степени (подразумевается как полустепень захода, так и полустепень выхода) графа не приводят к существенным изменениям в проценте неплатежеспособных банков:

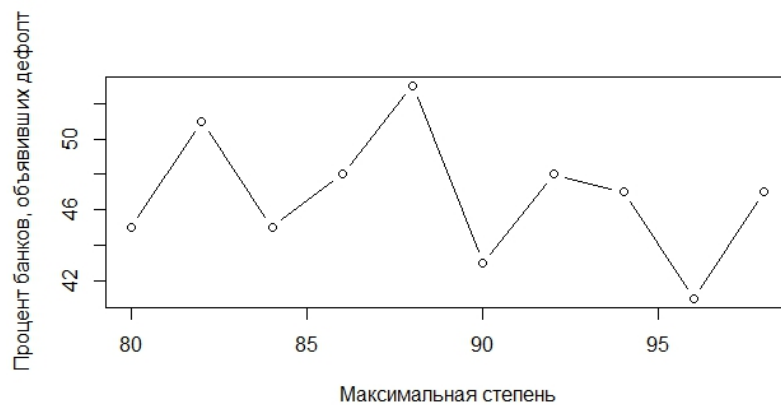


График 8

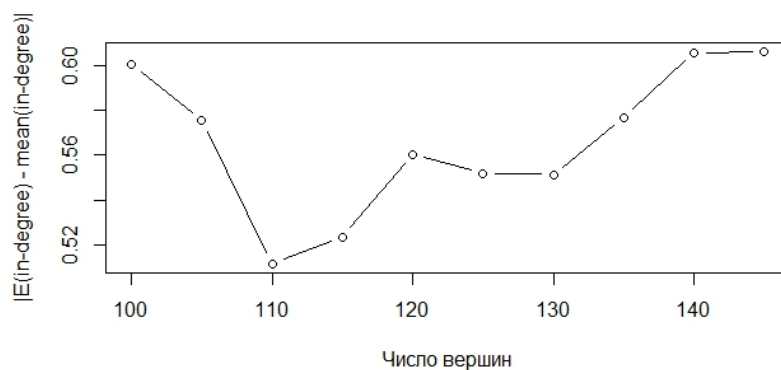


График 9

График 9 показывает, что независимо от количества вершин, мат. ожидание и среднее значение полустепени захода вершины при данном распределении почти совпадают.

Однако значения мат. ожидания и среднего значения полустепени выхода, напротив, стабильно различаются на достаточно большую величину:

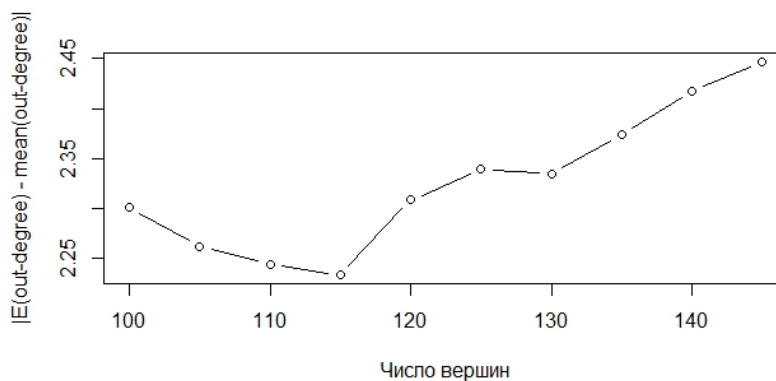


График 10

В следующих графиках размер выборки был увеличен до 100. Можно посчитать значение  $v_j$ , которое было выведено в (32), как отношение числа банков системы с полустепенью захода  $j$ , для которых выполнено (31), к общему числу банков с полустепенью захода  $j$ . В графиках представлены зависимости среднего числа различных полустепеней захода  $j$ , для которых  $v_j = 1$ . Причем при  $q = 1$  это число стабильно равно 2, а значения  $j$  при этом равны 4 и 5, т. е. это две минимально возможные полустепени захода. Аналогичные эффекты наблюдаются и для остальных значений  $q$ . Эти данные лишней раз иллюстрируют тот факт, что надежность системы укрепляется вместе с ростом полустепеней банков.

$q = 0.95:$

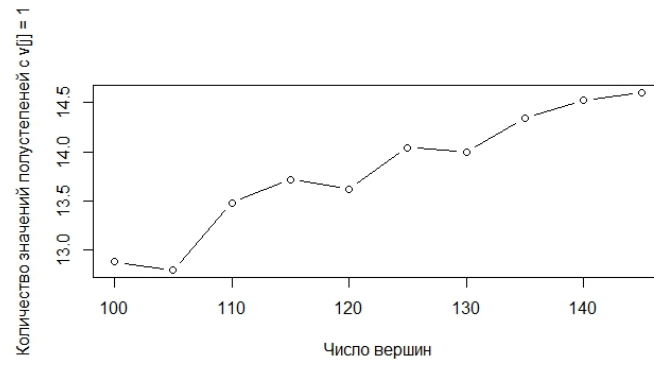


График 11

$q = 0.9:$

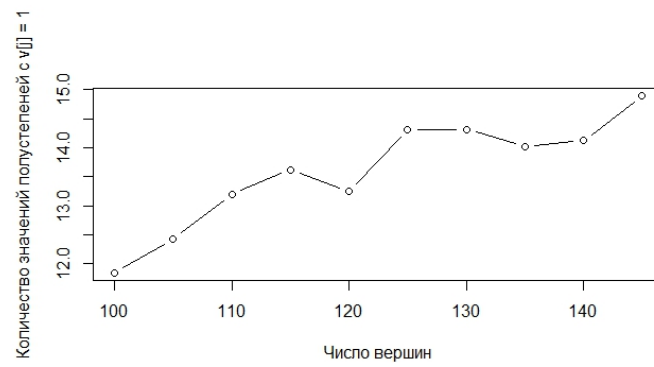


График 12

$q = 0.5:$

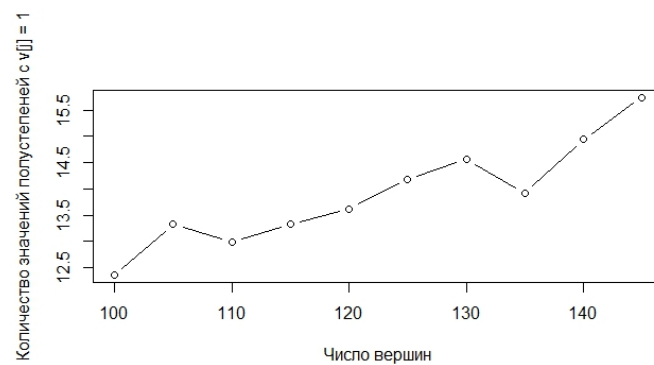


График 13

Кроме количества банков, которые объявляют дефолт, интересно посмотреть и среднюю глубину волны дефолтов (под глубиной подразумевается наибольшее расстояние от первого банка, объявившего дефолт, до некоторого другого банка, до которого дошла волна дефолтов):



График 14

Максимальные по выборке значения глубины дефолта:

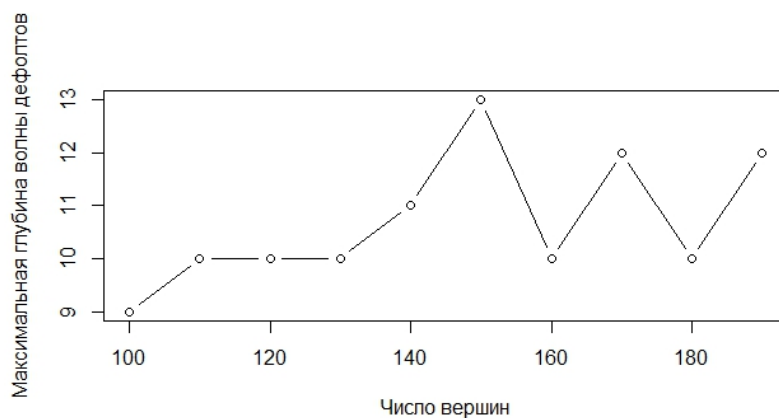


График 15



## Список литературы

- [1] M. Elliott, B. Golub, M. O. Jackson. *Financial Networks and Contagion*. American economic review, vol. 104, no. 10, 2014, pp. 3115-53.
- [2] Райгородский А. М. *Модели случайных графов*. М.: МЦНМО, 2011.
- [3] S. Kapadia, P. Gai. *Contagion in financial networks*. Proceedings of Royal Society Series A, 466(2120), 2010, pp. 2401–2423.
- [4] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. *Random graphs with arbitrary degree distributions and their applications*. Physical review E 64 (2), 026118, 2001, arXiv:cond-mat/0007235v2.
- [5] И. В. Романовский. *Моделирование случайных распределений. Конспект лекций по курсу "Дискретный анализ"*. [http://www.math.spbu.ru/ru/Archive/Courses/jvr/DA\\_html/\\_lec\\_1\\_07.html](http://www.math.spbu.ru/ru/Archive/Courses/jvr/DA_html/_lec_1_07.html)
- [6] A.V. Leonidov and E.L. Rumyantsev. *"Russian interbank networks: main characteristics and stability with respect to contagion"*, Proc. "Instabilities and Control of Excitable Networks: from macro- to nano- systems MIPT, 2012, arXiv: 1210.3814