

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

**Моделирование процессов диффузии в графах  
для анализа системных рисков сложных систем**

АВТОРЕФЕРАТ

студентки 2 курса 248 группы

направления 09.04.03 «Прикладная информатика»

механико-математический факультет

Мыльциной Анны Викторовны

Научный руководитель

Зав. кафедрой, д.ф.-м.н., доцент \_\_\_\_\_ С.П. Сидоров

Заведующий кафедрой

д.ф.-м.н., доцент \_\_\_\_\_ С.П. Сидоров

Саратов 2020

**ВВЕДЕНИЕ.** Сети повсеместны в социальных и экономических явлениях. Использование методов из теории графов позволило теории экономических сетей улучшить визуализацию и понимание тех экономических явлений, в которых нельзя пренебрегать принадлежностью индивидов к их социальным взаимоотношениям. Предполагается, что совокупное поведение экономики не может быть исследовано с точки зрения поведения изолированных индивидов, как это обычно делается в стандартной экономической теории. Фирмы взаимодействуют только с несколькими другими фирмами из всех присутствующих в экономике. Более того, существуют разные способы взаимодействия фирм, и они могут со временем научиться адаптировать свои взаимодействия, что означает, что они укрепляют прибыльные, а сокращают дорогостоящие отношения. Эта теория основывается на их предыдущем опыте. Затем можно рассматривать экономику как развивающуюся сеть.

Выпускная квалификационная работа посвящена изучению сетевых моделей в экономике, построению модельной сетевой экономической задачи, анализу построенной модели.

Актуальность работы подтверждается все большим интересом проявленным к сетевой экономике. Все чаще экономические сети строятся на основе теории графов.

Цели работы изучить построение экономических сетей и смоделировать взаимопроникновение и разрушение сети.

Для выполнения поставленных целей были решены следующие задачи:

1. Изучение языка программирования  $R$ , как языка позволяющего работать с большим объемом данных.
2. Собрать и проанализировать данные для построения модельной задачи. Данные должны удовлетворять требованиям полноты и однородности.
3. Построить модель сети.
4. Смоделировать рост и распад сети.

В первом разделе **Сети в экономике** приведена характеристика сетей и несколько примеров взаимодействий в сетях, динамика переменных состояний и эволюция сети. Один из примеров сети в виде двудольного графа, который может быть спроецирован на две однорежимные сети. Предположим, что на рисунке 1 каждый узел верхнего ряда, обозначенный цифрой, представляет совет директоров компании, а каждый узел, обозначенный буквой, представляет человека.

Пусть связь между человеком  $B$  и советом директоров 1 представляет тот факт, что человек  $B$  служит в совете 1. Но так же видно, что  $B$  служит и в совете 2.

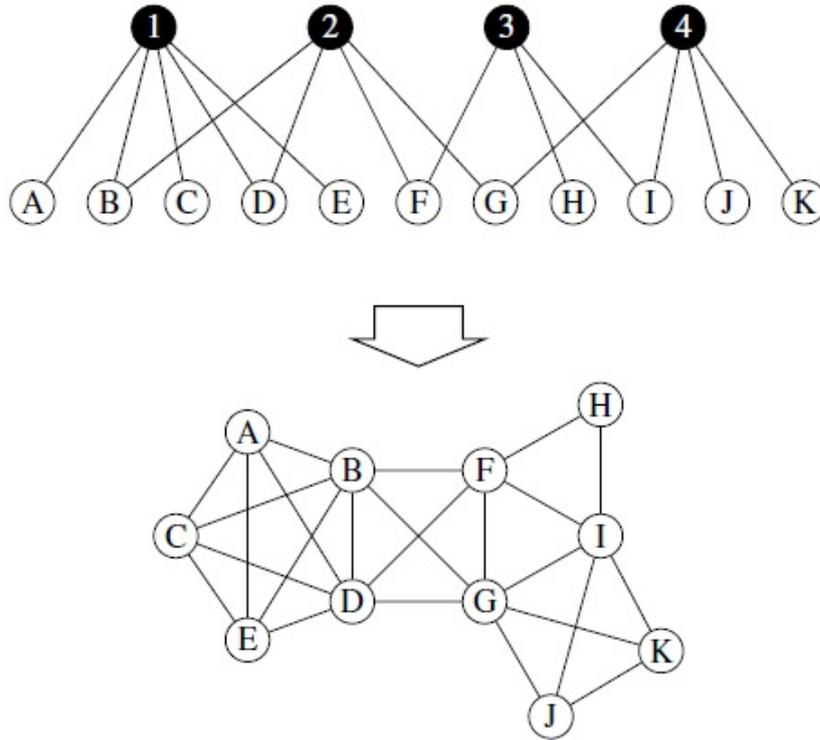


Рисунок 1 — Пример двудольной сети (вверху). Существует два класса узлов, и ссылки назначаются только между узлами, которые не принадлежат к одному и тому же классу. Однорежимная проекция - это новый граф, состоящий только из узлов одного класса, в котором связь между двумя узлами подразумевает, что в исходном двудольном графе два узла были связаны с одним и тем же третьим узлом

Однорежимная проекция на директоров - это новый граф, в котором есть связь между двумя людьми, если они работают вместе в одном или нескольких советах.

При выполнении этой проекции некоторая информация теряется. Ссылки не указывают, находится ли каждая пара директоров в разных советах или все три директора - в одном совете.

Так же в первом разделе рассмотрены элементы теории графов необходимые для описания сетей экономики. Изучены **меры сетей**:

1. Средняя длина пути  $L = \frac{1}{\frac{1}{2}n(n-1)} \sum_{i \geq j}^n d_{ij}$  - это среднее геодезическое (т.е. самое короткое) расстояние между парами узлов в графе, где  $d_{ij}$  - геодезическое расстояние от узла  $i$  до узла  $j$ .

2. Для каждого узла  $i$  коэффициент локальной кластеризации

$$C_l(i) = \frac{|\{e_{jk} \in E(G) : e_{ij} \in E(G) \wedge e_{ik} \in E(G)\}|}{d_i(d_i-1)/2}$$

и определяется как доля пар соседей  $i$ , которые сами являются соседями.

Глобальный коэффициент кластеризации имеет вид  $C_l = \frac{1}{n} \sum_{i=1}^n C_l(i)$ .

3. Центральность измеряет важность узла на основе его положения в сети.

- Степень центральности узла  $i$  - это просто количество ссылок  $d_i = \sum_{j=1}^n a_{ij} = \sum_{j=1}^n a_{ji}$  (так как  $A$  симметрична).
- Степень близость или центральность близости  $C_C(i) = \sum_{v \neq i} \frac{1}{d_{iv}}$  к  $i$  является обратной величиной суммы геодезических расстояний до всех других узлов графа.
- Промежуточная центральность для  $i$  узла равна  $C_B(i) = \sum_{u,v \neq i} \frac{g_{uv}(i)}{g_{uv}}$ , если  $g_{uv}$  число геодезических путей  $d_{uv}$  от  $u$  до  $v$  и  $g_{uv}(i)$  количество путей от  $u$  до  $v$ , проходящих через  $i$ .
- Центральность собственных векторов измеряет важность узла по важности его соседей.

Во втором разделе рассмотрены способы построения сетей - равномерной и с преимущественным типом распределения. Рассмотрены способы вычисления их распределения степеней.

В третьем разделе рассмотрены стратегии формирования сетей. Приведен теоретический материал, в котором доказывається эффективность и полезность разных видов сетей.

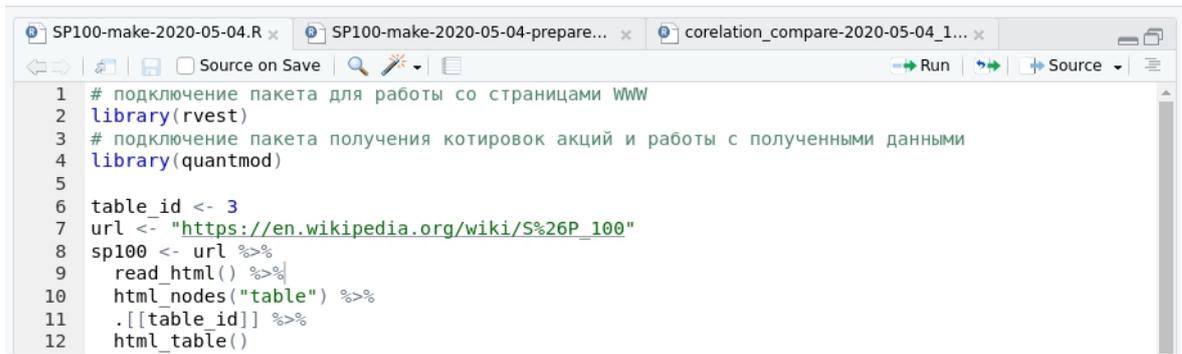
В четвертом разделе проведено **моделирование процесса роста и разрушения сети**. Для этого использовался язык программирования  $R$ , **как язык для работы с большими данными**. При написании программы для решения модельной задачи на языке  $R$ , были использованы пакеты `rvest`, `quantmod`, `igraph`.

Пакет `rvest` умеет одновременно и скачивать и обрабатывать веб-страницы. Скачивает с помощью `httr`, а обрабатывает `sxml2`.

Пакет `quantmod` используем для скачивания котировок (текстовые файлы CSV) и построения графиков.

На рисунке 2 приведен код программы на языке  $R$ , в котором осуществляется подключение пакета `rvest` и `quantmod`. В данном случае пакет `rvest` используется для получения списка тикеров (короткое название акций) и компаний, а пакет `quantmod` для получения котировок акций и работы с полученными данными.

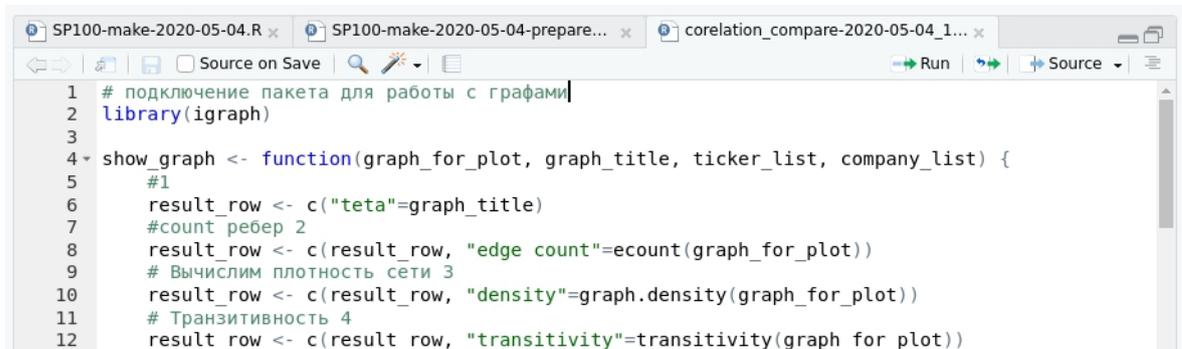
Следующий пакет в  $R$  `igraph` позволяет интерактивно строить графики сетей. `Igraph` позволяет выполнить построение парных сетей. У парных, или двудольных, графов встречаются два разных вида узлов и связей. Все ребра



```
1 # подключение пакета для работы со страницами WWW
2 library(rvest)
3 # подключение пакета получения котировок акций и работы с полученными данными
4 library(quantmod)
5
6 table_id <- 3
7 url <- "https://en.wikipedia.org/wiki/S%26P_100"
8 sp100 <- url %>%
9   read_html() %>%
10  html_nodes("table") %>%
11  .[[table_id]] %>%
12  html_table()
```

Рисунок 2 — Код программы на языке *R* - подключение пакетов *rvest* и *quantmod*

сети заданы в матричном виде. Их можно считать в объект-граф, используя `graph.incidence`. В *igraph* двудольные сети имеют параметр ребра `type`, равный 0 для одной группы вершин и 1 для другой.



```
1 # подключение пакета для работы с графами
2 library(igraph)
3
4 show_graph <- function(graph_for_plot, graph_title, ticker_list, company_list) {
5   #1
6   result_row <- c("teta"=graph_title)
7   #count ребер 2
8   result_row <- c(result_row, "edge count"=ecount(graph_for_plot))
9   # Вычислим плотность сети 3
10  result_row <- c(result_row, "density"=graph.density(graph_for_plot))
11  # Транзитивность 4
12  result_row <- c(result_row, "transitivity"=transitivity(graph_for_plot))
}
```

Рисунок 3 — Код программы на языке *R* - подключение пакета *igraph*

На рисунке 3 представлен код программы подключения пакета *igraph* на языке *R*. У *igraph* также есть специальное представление для двудольных сетей (хотя оно не всегда хорошо подходит, возможно, лучшим решением будет создать собственное двудольное представление), пример двудольной сети приведен на рисунке 1.

**Постановка задачи и обзор данных на которых происходит моделирование.** Возьмем котировки акций девяти компаний. В работе приведем код программы, в котором происходит подготовка данных к тому, чтобы их использовать для каких либо процедур. Например, выполнения с ними математических операций. По ним вычислим доходности и уже по доходностям подсчитаем корреляционную матрицу. На рисунке 4 приведен результат работы программы. Полученная корреляционная матрица отображает степень зависимости между ее параметрами. На рисунке 5 показано окно работы программы с выводом построенного графа. Результаты работы записаны в файлах. В приложении А содержатся коды трех файлов, в которых описан алгоритм работы. Файл 1

	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	1.0000000	0.2239046	0.2514108	0.0000000	0.0000000	0.2519011	0.0000000	0.2624235	0.3369029
2	0.2239046	1.0000000	0.3111430	0.2253164	0.2117696	0.2249066	0.0000000	0.2890665	0.3524935
3	0.2514108	0.3111430	1.0000000	0.3100829	0.0000000	0.2446948	0.0000000	0.5108979	0.5092172
4	0.0000000	0.2253164	0.3100829	1.0000000	0.0000000	0.2549779	0.0000000	0.2937451	0.3669989
5	0.0000000	0.2117696	0.0000000	0.0000000	1.0000000	0.0000000	0.2979342	0.0000000	0.2235759
6	0.2519011	0.2249066	0.2446948	0.2549779	0.0000000	1.0000000	0.0000000	0.2444120	0.3257936
7	0.0000000	0.0000000	0.0000000	0.0000000	0.2979342	0.0000000	1.0000000	0.0000000	0.2337564
8	0.2624235	0.2890665	0.5108979	0.2937451	0.0000000	0.2444120	0.0000000	1.0000000	0.6301738
9	0.3369029	0.3524935	0.5092172	0.3669989	0.2235759	0.3257936	0.2337564	0.6301738	1.0000000

Showing 1 to 9 of 9 entries, 9 total columns

Рисунок 4 — Корреляционная матрица

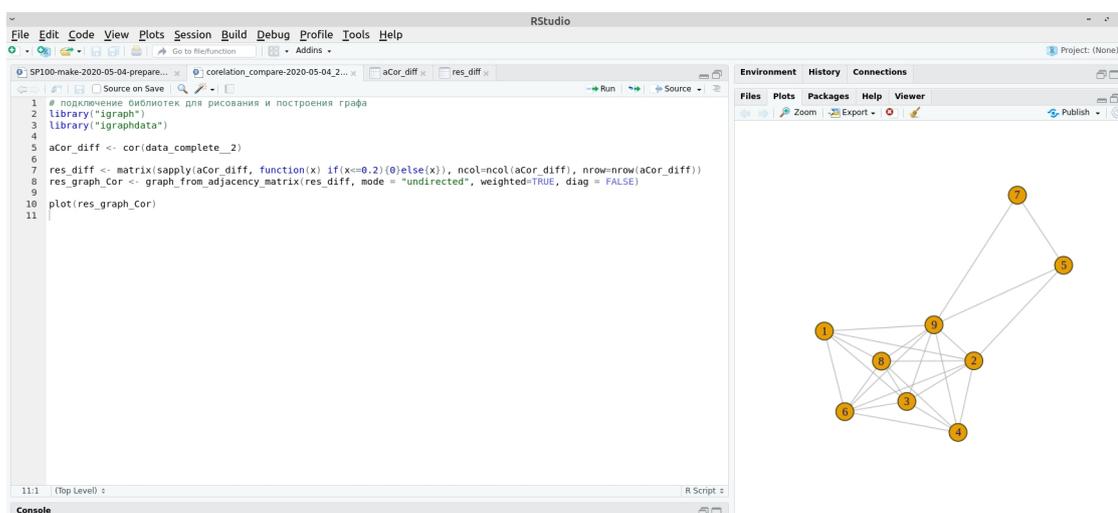


Рисунок 5 — Построение графа

- получение списка тикетов компаний и названий компаний
- исключение компании данных по которой не получается получить
- разделение общего списка 100 компаний на 2
- первый список из 90 компаний для которых есть данные с 2007 года
- второй список из 9 компаний, для которых есть данные с разных периодов
- оба списка записываем в два файла. В файлах список с историческими данными, список тикетов и список названий

## Файл 2

- при каждом запуске мы будем получать разные последовательности данных
- открытие файла с данными полученным на первом этапе

- создаём список случайный последовательностью
- по созданному списку перестраиваем данные из файла
- используя пакет финансовых вычислений, извлекаем приведённую цену закрытия
- по приведённой цене вычисляем ежедневную доходность по логарифмическому закону
- полученные данные оставляем в памяти, для дальнейшего использования
- не используемые переменные удаляем из памяти

### Файл 3

- функция для расчёта параметров графа и вывод его на экран
- на вход функции подаётся граф, список тикетов, список компаний, матрица на основе которой был создан граф
- отбираем из первого списка из 90 компаний, нужное нам количество для исследований
- по отобранным данным, строим корреляционную матрицу
- также строим корреляционную матрицу для второго списка компаний
- запускаем цикл для исследований по уровням отсечений
- отсечения идут от 0,2 до 0,9
- по заданному уровню отсечения перестраиваем матрицу корреляций по правилу, если данные в ячейки матрицы меньше или равно уровню отсечения, то ставим 0, иначе 1
- результат преобразования матрицы корреляций получается матрица связей между компаниями
- полученная матрица связей используется для построения графа
- полученный граф и данные по которым он был построен передаём в функцию и получаем результирующую строку для отчетного списка

Результаты работы описываем фазами.

Фаза 1

- используем данные из списка компаний 1
- получаем компанию с самым большим количеством связей

#### Фаза 2

- исключаем из матрицы связей компанию с самым большим количеством связей
- исследуем данный вариант

Строим разницу между результатами фазы 1 и фазы 2.

#### Фаза 3

- используем данные из списка компаний 2
- получаем компанию с самым большим количеством связей

#### Фаза 4

- создаём список компаний из списка компаний 1 и компании из второго списка с самым большим количеством связей
- в полученном списке исключаем из матрицы связей компанию полученную в фазе 1

Полученные в результате исследований результаты сохраняем в файл CSV

**Результаты** работы программы содержатся в приложении Б.

Визуализация работы программы (файлы 1-3) по фазам 1-4 отражена на рисунках 6 – 9.

На рисунках 6 – 9 красным выделены вершины имеющие самое большое число связей. Рассмотрим рисунки соответствующие  $\theta = 0,3$ . После прохождения всех фаз получаем, что бывают ситуации когда вновь добавленная компания становится компанией с максимальным количеством связей (пример построен на тридцати и девяти компаниях).

**ЗАКЛЮЧЕНИЕ.** Сетевая экономика – это среда, в которой любая компания или индивид, находящийся в любой точке экономической системы, могут с помощью интернет-технологий контактировать легко и с минимальными затратами с любой другой компанией или индивидом по поводу совместной работы, для торговли, для обмена идеями или просто для удовольствия.

Все чаще для представления сетевой экономики пользуются формулой: сетевая экономика = традиционная экономика + информационные ресурсы и технологии. В настоящее время к сетевой экономике тесно примыкают те



0.5

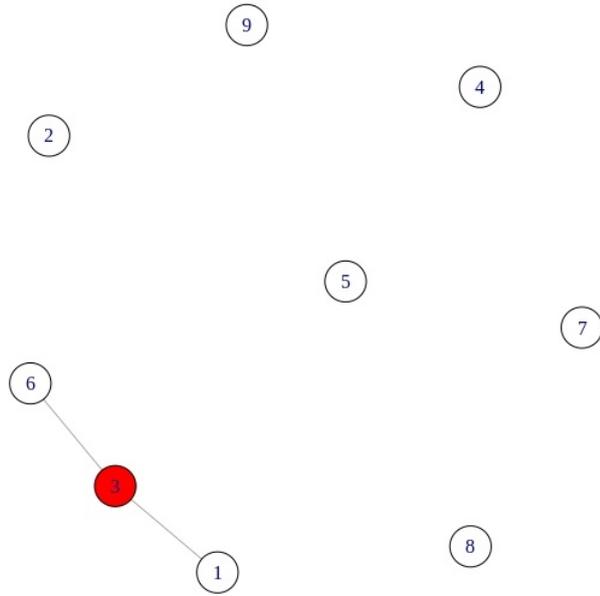


Рисунок 8 — Фаза 3  $\theta = 0,5$

0.5

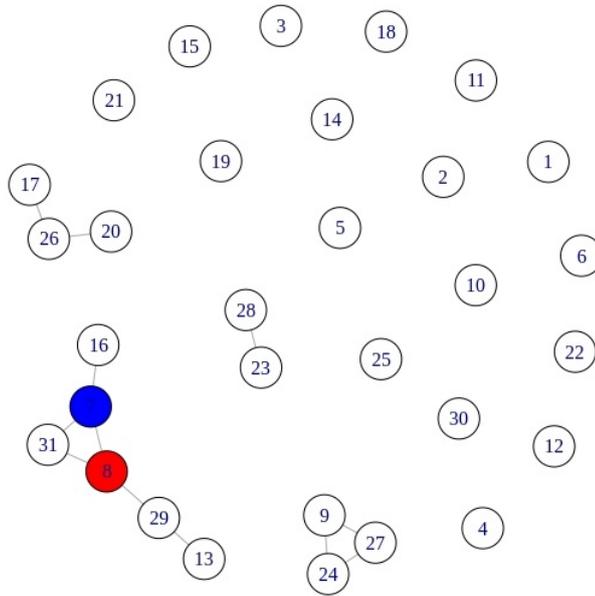


Рисунок 9 — Фаза 4  $\theta = 0,5$

помощи пакетов языка *R* были получены временные данные, отобраны данные для анализа, построена корреляционная матрица. С помощью пакета анализа графов построены графы по полученной корреляционной матрице. Графы проанализированы, то есть вычислены число ребер, вершин и др., и визуализированы.